

Screenwriting 2.0

What are the possibilities of screenplay 'datafication'?

How the screenplay as data can impact creating and managing, presenting and sharing, analyzing and visualizing textual screenplay content.

Stewart McKie

PhD Degree

Media Arts

Royal Holloway College, London

Declaration of Authorship

I, Stewart McKie, hereby declare that this thesis and the work presented in it is entirely my own. Where I have referenced or consulted the work of others, this is always clearly stated whether in the text of the thesis or on the credits page of the Scenepad online application software deliverable.

Signed: 

Date: 09 August 2013

Abstract

Screenwriting software applications have traditionally focused on the screenplay as document, essentially enabling writers to create screenplays in an industry standard format that can be printed or distributed electronically. This thesis explores the possibilities of technology delivering more value if the screenplay content is stored at a more granular level, as data and metadata. This 'datafication' of the screenplay content enables new and interesting possibilities for screenwriting practice.

Managing screenplay content as data rather than 'document' better reflects the dynamic nature of the screenplay and collaborative nature of screenwriting as a social and business process. Screenplay as data also facilitates automated analysis and visualization ('analytics') of screenplay content that can benefit both screenwriters and other stakeholders in the screenwriting process. This thesis focuses specifically on the topics of screenplay as data, the screenplay as social network and screenplay analytics.

The practical submissions that complement this thesis comprise a number of online application prototypes that explore some of the propositions discussed in the thesis argument. My final practical submission - Scenepad - is a complete online screenwriting application that is informed by the various prototypes that preceded it. Scenepad puts many of my thesis propositions into practice to deliver an evolutionary development in screenwriting technology I call 'Screenwriting 2.0'. I conclude this thesis by outlining some possible future developments of the screenplay as artefact, and screenwriting as practice.

Contents

DECLARATION OF AUTHORSHIP	2
ABSTRACT	3
CONTENTS	5
FIGURES AND TABLES	8
ACKNOWLEDGEMENTS	13
SOFTWARE DELIVERABLES	15
INTRODUCTION	16
FOCUS: THE FEATURE FILM SCREENPLAY	18
DATAFICATION	21
RESEARCH QUESTION	23
SCREENWRITING 2.0	25
SOURCES	26
METHODOLOGY	28
1 SCREENPLAY AS DATA	52
1.1 THE DATA-CENTRIC SCREENPLAY	54
1.2 DATA AND METADATA	55
1.3 UNSTRUCTURED VS. STRUCTURED DATA	57
1.4 TABULATED SCREENPLAYS.....	64
1.5 DOCUMENTED SCREENPLAYS.....	66
1.6 TAGGED SCREENPLAYS.....	68
1.7 XBRL AND CORPORATE REPORTING.....	72
1.8 ANALYTIC METADATA FOR SCREENWRITERS.....	76
1.9 SCREENPLAY ELEMENTS	78
1.10 SCREENPLAY DECOMPOSITION	84
1.11 SCREENPLAYS AS 'BIG DATA'.....	90
1.12 SCREENPLAY APIS.....	93
1.13 SCREENPLAY AS DATA WALL OR STREAM	95
1.14 SCREENPLAY AS INTERFACE	95
SUMMARY	96
2 SCREENPLAY ANALYTICS	98

2.1 WHY BOTHER ANALYZING SCREENPLAY CONTENT?	106
2.2 WHAT IS WORTH ANALYZING?	116
2.3 CHARACTER	119
2.4 NARRATIVE.....	133
2.5 STRUCTURE.....	140
2.6 ANALYTIC ARCHETYPES.....	144
2.7 GENERIC DATA ANALYSIS AND VISUALIZATION TOOLS.....	153
2.8 USING 'BIG DATA' TOOLS FOR SCREENPLAY ANALYSIS	165
SUMMARY	173
3 SCREENPLAY AS SOCIAL NETWORK	175
3.1 FUNCTIONALITY OF ONLINE SOCIAL NETWORKING	178
3.2 SCREENPLAY STAKEHOLDERS	182
3.3 SCREENWRITING AS BUSINESS PROCESS	187
3.4 SCREENPLAY AS SOCIAL NETWORK ANCHOR	196
SUMMARY	198
4 SCREENPLAY AS DESIGN.....	199
4.1 DATA-DRIVEN SCREENPLAY MANUFACTURING.....	202
4.2 SCREENPLAY MODELS	208
4.3 SCREENPLAY PATTERN LANGUAGE	210
4.4 REVERSE-ENGINEERING SCREENPLAY DESIGN	219
4.5 EVENT INDICATORS	224
SUMMARY	227
5. THE ROAD TO SCENEPAD.....	228
5.1 SCREENPLAY ANALYTICS TODAY.....	231
5.2 SOCIAL SCREENWRITING TODAY.....	243
5.3 SCREENWRITING 2.0 PROTOTYPES	244
5.4 SCRIPTCLOUD [2006].....	246
5.5 CONTENTCLOUD [2008]	253
5.6 SCRIPTGEIST [2007]	256
5.7 SCENECLASS [2010]	261
SUMMARY	266
6 SCENEPAD	267
6.1 SCENEPAD – SCREENPLAY AS DATA	268
6.2 SCENEPAD - SCENEWRITING	271

6.3 SCENEPAD AS SOCIAL NETWORK	278
6.4 SCENEPAD ANALYTICS.....	289
6.5 SCENEPAD FOR BUSINESS	298
6.6 SCREENPLAY PRODUCTION.....	299
SUMMARY	300
7. WHAT HAPPENS NEXT?	302
BIBLIOGRAPHY	309
APPENDIX A: HOFFMAN’S HELLO WORLD XBRL EXAMPLE	335
APPENDIX B: SELECTIVE SCENEPAD FEATURE LIST	338
APPENDIX C: SCREENPLAY ANALYTICS – A USE CASE.....	340
1. ON-SCREEN TIME	343
2. DIALOG AND ACTION	346
3. WHAT THE PROTAGONIST SAYS AND DOES	347
4. PROTAGONIST & ANTAGONIST ROLE METADATA.....	349
5. PROTAGONIST SOCIAL NETWORK.....	351
6. PROTAGONIST SENTIMENT	352
7. PROTAGONIST VS. ANTAGONIST.....	353
CONCLUSION	356

Figures and Tables

FIGURE 1.1 - GOOGLE NGRAM VIEWER PLOTS SPECIFIC NAME MENTIONS OVER TIME.....	22
FIGURE 1.2 – COMMENT POSTED BY ALEX EPSTEIN ON HIS <i>COMPLICATIONS ENSUE</i> BLOG.....	38
FIGURE 1.3 – COMMENT POSTED BY A STAGE32.COM USER ON THEIR BLOG AND SENT TO ME BY EMAIL ON 31 DECEMBER 2013.....	39
TABLE 1.1 – FUNCTIONAL EVOLUTION OF PROTOTYPE DELIVERABLES.....	51
TABLE 1.2 – TEXT FILE REPRESENTATIONS.....	59
TABLE 1.3 - ROW/COLUMN INTERSECTION.....	62
FIGURE 1.2 - SCRIPT AS TABLES (SIMPLIFIED).....	65
FIGURE 1.3 - SCRIPT AS DOCUMENTS (SIMPLIFIED).....	68
FIGURE 1.4 - AN EXAMPLE OF A SCREENPLAY HIERARCHY MODEL.....	71
FIGURE 1.5 – TYPICAL SCREENPLAY ELEMENTS.....	80
FIGURE 1.6 - CHATMAN’S NARRATIVE ENTITY CLASSES.....	87
FIGURE 1.7 – QUALITY PYRAMID OF CROWDSOURCED DATA.....	93
FIGURE 2.1 - FREYTAG’S PYRAMID.....	101
FIGURE 2.3 - WATCHING THE SCRIPT.....	102
FIGURE 2.4 - BRANDWATCH SOCIAL OSCARS DATAVIZ.....	107
FIGURE 2.5A – STORY CHART FOR CASABLANCA.....	115
FIGURE 2.5B – MCKEE’S STORY CHART OF PULP FICTION.....	116
TABLE 2.1 - POTENTIAL SCREENPLAY ANALYSIS TOPICS.....	117
FIGURE 2.6 - THE REFORMER ENNEAGRAM STYLE.....	123
FIGURE 2.7 - PROTAGONIST-DRIVEN CAST DESIGN (MCKEE 1998:380).....	125
FIGURE 2.8 - CHARACTER CONFLICT AS STORY (ADAPTED FROM BABOULENE 2012:20).....	125
FIGURE 2.9 – SOCIAL NETWORK OF HAMLET (SCENE VI).....	127
FIGURE 2.10 – CHARACTER WEIGHTS IN <i>BUFFY</i>	128
FIGURE 2.11 - DYNAMIC CHARACTER MATRIX.....	133
FIGURE 2.12 - PULP FICTION NARRATIVE LINES.....	138
FIGURE 2.13 - FLASHBACK AS THWARTED DREAM: SHINE.....	139
FIGURE 2.14 - FIELD’S THREE-ACT STRUCTURE.....	146
FIGURE 2.15 - CONVERGING PLOT LINES IN <i>THE DAY OF THE JACKAL</i> (POPE, 1998:173).....	147
FIGURE 2.16 - VOGLER’S HERO’S JOURNEY (VOGLER, 1994:194).....	148
FIGURE 2.17 - MURDOCK’S HEROINE’S JOURNEY (MURDOCK, 1990:5).....	148
FIGURE 2.18 - <i>DIE HARD</i> ANALYSIS (VOYTILLA, 1999:41).....	149
FIGURE 2.19A - ROLLERCOASTERS OF ESCALATING TENSION (COOPER, 1997:71).....	150
FIGURE 2.19B - PAUL AND JACK’S FRACTURED TANDEM IN <i>21 GRAMS</i> (ARONSON, 2010:293).....	151
FIGURE 2.20 - SCREENPLAY STRUCTURE AS NODE TREE (KITCHEN, 2006:162).....	152

FIGURE 2.21 - CHARACTERS IN THE THIRD MAN (MACKENDRICK, 2005:19).....	153
FIGURE 2.22 - ARC DIAGRAM APPLIED TO MY SCREENPLAY CARPINI.....	155
FIGURE 2.23- A TOPIC FLOWER APPLIED TO A SCREENPLAY.....	157
FIGURE 2.24 - A SHARED WORD DIAGRAM APPLIED TO TWO DRAFTS OF A SCREENPLAY.....	158
FIGURE 2.25 - NAMING NAMES.....	159
FIGURE 2.26 - TRANSCRIPT ANALYZER.....	160
FIGURE 2.27 – TRANSCRIPT ANALYSIS.....	161
FIGURE 2.28 – ARGO WORD CLOUD.....	163
FIGURE 2.29A - STARKY AND HUTCH STORYLINES.....	164
FIGURE 2.29B - HILL STREET BLUES STORYLINES.....	164
FIGURE 2.29.C - THE SOPRANOS STORYLINES.....	165
FIGURE 2.30 – SPLUNK SNIPPET ANALYSIS.....	166
FIGURE 2.31 – SPLUNK WORD ANALYSIS.....	166
FIGURE 2.32 – SPLUNK ROLE ANALYSIS.....	167
FIGURE 2.33 – SPLUNK LOCATION ANALYSIS.....	168
FIGURE 2.34 – BIGML SCRIPT ANALYSIS.....	169
FIGURE 2.35 – BIGML SCENE PREDICTION.....	170
FIGURE 2.36 – BIGML GENDER PREDICTION.....	171
FIGURE 2.37 – DATASEED ANALYSIS.....	172
FIGURE 2.38 – SAMPLE OF UPLOADED DATA FILE.....	173
TABLE 3.1 – SOCIAL NETWORK CHARACTERISTICS.....	177
FIGURE 3.1 - THE 3 PRINCIPAL SCREENPLAY-ANCHORED SOCIAL NETWORKS.....	178
FIGURE 3.2A - WRITE – REFLECT - REWRITE.....	191
FIGURE 3.2B - THE ENRICHED SCREENWRITING PROCESS.....	192
FIGURE 4.1 – IAN BARRY’S SCRIPT BOOK.....	200
FIGURE 4.2 - A LINED SCRIPT IN SCRIPT SUPERVISOR.....	201
FIGURE 4.3 - SCREENPLAY AS ADDITIVE LAYERS.....	206
FIGURE 4.4 – GAULD’S STRUCTURAL JOURNEY.....	206
FIGURE 4.5 – YORKE’S PARADIGM SUMMARY.....	209
TABLE 4.1 – PATTERN ATTRIBUTES.....	212
TABLE 4.2 – 3-ACT STRUCTURE PATTERN.....	213
TABLE 4.3 – PROTAGONIST PATTERN.....	215
FIGURE 4.6 – ALIEN AS VISUALIZED BY MILESI & MATTEO (2012:9).....	218
FIGURE 4.7 - INSPECTOR MONTALBANO INFOGRAPHIC.....	219
FIGURE 4.8 - STOVIZ STORYLINE VISUALIZER.....	221
FIGURE 4.9 – CHUN’S STORY VISUALIZER.....	223
FIGURE 5.1 - WORKING WITH A FORMATTED SCRIPT IN CELTX.....	232
FIGURE 5.2 - NAVIGATING A SCRIPT HIERARCHY IN SOPHOCLES.....	233
FIGURE 5.3 - MANAGING INDEX CARDS IN SUPERNOTECARD.....	234

FIGURE 5.4 - SCRIPT ANALYSIS CHARTS IN SOPHOCLES	235
FIGURE 5.5 - THE SOCIAL NETWORK OF CASABLANCA IN SOPHOCLES	237
FIGURE 5.6 – PLOTBUILDER EXCITEMENT GRAPH	238
FIGURE 5.7 - THE STORYVIEW STRUCTURAL VIEW OF PULP FICTION.....	239
FIGURE 5.8 - THE STORYVIEW SCRIPT WALL POSTER.....	239
FIGURE 5.9 - A POSSIBLE TIMELINE, STRUCTURE & THROUGHLINE VISUALIZATION	240
FIGURE 5.10 - SCRIPT STORYBOARD IN CELTX	241
FIGURE 5.11 – AMAZON STORYTELLER STORYBOARD.....	242
TABLE 5.1 – LIST OF PROTOTYPES.....	246
FIGURE 5.12 – SCRIPTCLOUD STATISTICS JULY 6, 2008	248
FIGURE 5.13 – SCRIPTCLOUD.COM HOME PAGE	248
FIGURE 5.14 - SCRIPTCLOUD OF MY OWN <i>CARPINI</i> SCRIPT	250
FIGURE 5.15 - <i>THE GODFATHER</i> SCRIPTCLOUD.....	252
FIGURE 5.16 - <i>THE GODFATHER</i> KEYWORDS	252
FIGURE 5.17 - <i>THE FRENCH CONNECTION</i> SCRIPTCLOUD	253
FIGURE 5.18 - CONTENTCLOUDS HOME PAGE	254
FIGURE 5.19 - COMPARING GODFATHER SCRIPTS	255
FIGURE 5.20 - COMPARING ACTS IN SHAKESPEARE’S <i>MACBETH</i>	256
FIGURE 5.21 - SCRIPTGEIST HOME PAGE.....	257
FIGURE 5.22 - SCRIPTGEIST STATISTICS.....	258
FIGURE 5.23 - <i>THE GODFATHER</i> DIALOG CLOUD.....	259
FIGURE 5.24 - <i>THE GODFATHER</i> CHARACTER THROUGHLINE.....	259
FIGURE 5.25 - <i>THE GODFATHER</i> INT/EXT SCENES.....	260
FIGURE 5.26 - <i>THE GODFATHER</i> CHARACTERS AND PART SIZE.....	260
FIGURE 5.27 - <i>THE GODFATHER</i> CHARACTER DIALOG	261
FIGURE 5.28 - SCENECLASS HOME PAGE.....	262
FIGURE 5.29 – SCENECLASS CLASS ENTITY	263
FIGURE 5.30 - SCENECLASS SCENE ENTITIES	263
FIGURE 5.31 – SCENECLASS SCENE COMMENTS	264
FIGURE 5.32 – SCENECLASS STUDENT RESPONSES	265
FIGURE 6.1 - ORIGINAL <i>SCENEPAD1</i> BETA HOME PAGE	268
FIGURE 6.2 – CORE SCENEPAD DATA ENTITIES (TABLES)	270
FIGURE 6.3 - MARINER <i>STORYMILL</i> SCENE SNAPSHOTS.....	273
FIGURE 6.4 - <i>SCENEPAD1</i> SCENE VERSIONING.....	274
FIGURE 6.5 – <i>SCENEPAD3</i> – SCENE CONTENT TYPES AND ANALYSIS	275
FIGURE 6.6 – <i>SCENEPAD4</i> MANAGING SCENE GROUPING	275
FIGURE 6.7 - <i>SCENEPAD3</i> HERO’S JOURNEY TEMPLATE	276
FIGURE 6.8 – <i>SCENEPAD1</i> ACT AND SEQUENCE NAVIGATOR	277
FIGURE 6.9 – <i>SCENEPAD4</i> SCENE STORYLINE MANAGER	277

FIGURE 6.10 – SCENEPAD4 SCENE STORYBOARD	278
FIGURE 6.11 – SCENEPAD4 GROUPS.....	279
FIGURE 6.12 - SCENEPAD4 GROUP USERS.....	280
FIGURE 6.13 - SCENEPAD4 SCRIPT MANAGER CONTENT HUB	282
FIGURE 6.14 - SCENEPAD4 ROLE GALLERY.....	282
FIGURE 6.15 – SCENEPAD4 SCENE COMMENTING	284
FIGURE 6.16 - SCENEPAD4 SCRIPT QR CODE.....	285
FIGURE 6.17 - SCENEPAD4 MOBILE SITE HOME PAGE	286
FIGURE 6.18 - SCENEPAD2 - SHOWING TUMBLR-SOURCED IMAGES DISPLAYED ON SCENEPAD ROLE WALL	287
FIGURE 6.19 - SCENEPAD4 - SHOWING IMAGES SOURCED FROM KEE.PS DISPLAYED UNDER SCRIPT KEEPS TAB.....	288
FIGURE 6.20 - SCENEPAD4 - SHOWING TWEETS SOURCED FROM TWITTER DISPLAYED UNDER SCRIPT TWEETS TAB.....	288
FIGURE 6.21 - SCENEPAD1 SCRIPT DASHBOARD.....	290
FIGURE 6.22 - SCENEPAD1 SCENE DASHBOARD.....	290
FIGURE 6.23 - SCENEPAD4 ROLE-DIALOG.....	291
FIGURE 6.24 - SCENEPAD1 TOP 10 ROLES	292
FIGURE 6.25 - SCENEPAD1 PROTAGONIST VS. ANTAGONIST	292
FIGURE 6.26 - SCENEPAD1 TOP 10 LOCATIONS.....	293
FIGURE 6.27 - SCENEPAD1 SCRIPT-LEVEL WORD CLOUDS.....	294
FIGURE 6.28 - SCENEPAD3 - ALIEN KEYROLE THRULINE.....	296
FIGURE 6.29 - SCENEPAD3 ALIEN END OF SCRIPT THRULINE	296
FIGURE 6.30 - SCENEPAD4 DIALOG SENTIMENT ANALYSIS	297
FIGURE 6.31 – SCENEPAD4 MANAGING SCRIPT COVERAGE	298
FIGURE 6.32 - SCENEPAD4 SCRIPT OWNERSHIP DASHBOARD	299
FIGURE 6.33 - SCENEPAD4 SCRIPT DAY CONTENT HUB.....	300
FIGURE C1 – SCRIPT HOME STATISTICS.....	341
FIGURE C2 – SCRIPT ROLES	342
FIGURE C3 – ROLE HOME STATISTICS	342
FIGURE C4 – ROLE FAQs	343
FIGURE C5 – TOP 5 ROLES BY SCENE (NOTE: EACH DOT REPRESENTS A UNIQUE SCENE)	344
FIGURE C7 – TOP 5 ROLES BY EXIT SCENE (NOTE: EACH DOT REPRESENTS A UNIQUE SCENE).....	346
FIGURE C8 – TOP 5 ROLES BY DIALOG SNIPPETS (NOTE: EACH DOT REPRESENTS A UNIQUE DIALOG SNIPPET)	347
FIGURE C9 – TOP 5 ROLE CUES MENTIONED IN ACTION SNIPPETS	347
FIGURE C10 - MOST FREQUENT NOUNS/WORDS IN DIALOG SNIPPETS SPOKEN BY ROLE	349
FIGURE C11 - MOST FREQUENT NOUNS/WORDS IN ACTION SNIPPETS OF ROLE	349
FIGURE C12 – BASIC ROLE METADATA.....	350
FIGURE C13 – KEY ROLE FAQs.....	351

FIGURE C15 – MAGGIE’S SENTIMENT ANALYSIS.....	352
FIGURE C16 - SCRIPT BALANCE: PROTAGONIST VS. ANTAGONIST DIALOG SCENES	353
FIGURE C17 - SCRIPT BALANCE: PROTAGONIST VS. ANTAGONIST DIALOG SNIPPETS.....	354
FIGURE C18 - SCRIPT BALANCE: PROTAGONIST VS. OTHER CAST SCENES.....	355
FIGURE C19 – DOREEN (ANTAGONIST) AS SOCIAL NETWORK HUB	355
FIGURE C20 – DIALOG GENDER BALANCE	356

Acknowledgements

I would like to acknowledge the contribution of the following people in relation to my PhD research.

Department of Media Arts, Royal Holloway

The Department of Media Arts provided a tuition fee waiver for the duration of my studies at Royal Holloway.

Supervisors

My Royal Holloway, London supervisors Adam Ganz and Fionn Murtagh for their support and help over the duration of my research.

Coders and Application Component Providers

All the individuals and companies listed in the Scenepad Credits page (see <http://scenepad4.tripos.biz/credits.php>)

Scenepad Beta-Testers

The beta-test group of academics and students, especially Carmen Sofia Brenes of the University of the Andes, who agreed to try out the Scenepad prototype and the hundreds of other users who signed up to try out my other online application prototypes.

LCACE Funding

London Centre for Arts and Cultural Exchange (LCACE) for funding the Scriptgeist prototype delivered in 2008 based on an application made by Lydia Daniels of the Royal Holloway Enterprise center and supported by Adam Ganz.

Conferences

The Semantic Media and Adaptation and Personalization (SMAP) conference

for allowing me to present my paper *Scriptcloud.com: Content Clouds for Screenplays* at their doctoral consortium in 2007.

The Screenwriting Research Network (SRN) for allowing me to make presentations at both their Leeds (2008) and Copenhagen (2010) conferences.

Bournemouth University and the Southern Script Festival for allowing me to present my SceneClass and Scenewrite software prototypes at their 2011 conference.

Others

Sue Clayton and Adam Ganz of Royal Holloway and the MA students who made use of the initial prototype of my Wordpress SceneClass plugin during one of their residential program classes.

The Faculty and students of the department of Media Arts at Royal Holloway for their interest in my work at the annual PhD 'Awayday' presentations.

Students of Shaftesbury School in Dorset, who worked with me to produce a series of short films from an after-school scriptwriting class I ran, funded by the Shaftesbury School Foundation.

John Foster of Bournemouth University who allowed me to present my Scenewrite prototype to his MA Screenwriting students and discuss it with them.

Karl Phillips of University College Falmouth who allowed me to present to his BA Filmmaking students on the subject of screenplay analytics and provided £500 in seed funding for the development of Scenepad.

Software Deliverables

The assessment deliverable, Scenepad, is an online software application for screenwriters and screenplay management that applies many of the propositions of this thesis into practice.

Scenepad is located at <http://scenepad4.tripos.biz>

A second deliverable, ScriptFAQ, focused purely on screenplay analytics was completed during my PhD 'modification' period and can be found here:

<http://scriptfaq.tripos.biz>

The generic assessment login for both applications is:

Username rhul@scenepad.com

Password mckiephd2013

To review other software prototypes see 'Road to Scenepad' section.

Introduction

In *What Happens Next. A History of American Screenwriting*, Marc Norman ends with a statement emphasizing the uncertain impact of new technologies on the practice of screenwriting:

It's pointless to spend much time speculating on what lies in the future for Hollywood and its screenwriters, the result of new technologies and technologies not yet imagined on the narratives... It's a question of what happens next. (2008:485)

In a similar vein, Steven Maras in *Screenwriting: History, Theory and Practice* also recognizes challenging times ahead for the practice of screenwriting:

Today, screenwriting seems on the brink of a significant transformation. While digital technology and changes in film-making techniques may prompt reflection about what writing for the screen means, established ways of thinking about screenwriting continue to structure or 'haunt' debates about screenwriting, and perhaps limit a more pluralistic understanding of writing. (2009:186):

And in *The screenplay as prototype*, Millard recognizes that:

The rise of new technologies and networks means that writing now happens primarily in digital environments" and "new conceptualizations of writing suggest a more fluid set of processes than traditional models of script development employ or allow. (2011:143-144)

These and other recent commentators on the screenplay and the practice of screenwriting recognize that technology can and is having an impact both on our concept of the screenplay as an artefact and the process of screenwriting as a practice. But in reality, while new technologies are revolutionizing the production, editing, special effects and supply-chain distribution of digital movies for example – screenplays and screenwriting seem to be stuck in a technology time warp.

I propose that a key reason for this 'stuckness' is the failure of screenwriting practice to move away from screenplay as document and towards screenplay as data, a transition that is happening in other areas of business

content management. For example in section 1.7 below, I refer specifically to corporate annual reporting, where the transition from submission of digitized documents (e.g. in HTML or PDF formats) to documents decomposed into machine-readable data points is gaining acceptance and delivering benefits both to producers and consumers of this content.

The document-centric bias of screenwriting is also evident in screenwriting technology (i.e. screenwriting software applications used by practicing screenwriters for writing screenplays). Over the last couple of decades, since its emergence in the 1980s, the functional focus of screenwriting technology has been on a specialist form of word processing for screenwriters. By which I mean providing the functionality for a screenwriter to use a software application specifically to write and revise a script, to format it for printing according to industry-standard formatting conventions, to view and navigate the content, and to print and export that content. Or put another way, a focus on the 'digitization' of screenplays into digital documents.

Over time, this initial functional focus of screenwriting applications on writing, formatting, viewing and publishing screenplay content, has been supplemented by new application capabilities to support both 'upstream' and 'downstream' screenwriting process activities. Upstream activities include story research, planning and outlining; downstream activities include linking storyboards to the script, and generating movie pre-visualizations or production breakdown reporting directly from the script content.

However there has been minimal attention paid to treating and storing screenplay content as data even with the release of many new screenwriting applications during the last decade. Over the 7 year period I have been researching, the number of screenwriting applications available to buy or available for free has at least doubled and many screenwriting applications are now delivered in new ways, that leverage new form factors and devices (i.e. the smartphone and tablet).

Certainly the widespread availability of a digital screenplay form (notably Adobe's PDF file format) has already changed the way screenplays are viewed, navigated, searched, submitted to agents and competitions and distributed to workgroups. And the availability of screenwriting software on new devices and form factors has changed how, when and where screenplays can be written and managed. But fundamentally we are still far from any kind of paradigm-shift to something that might truly be considered 'Screenwriting 2.0' (to adapt the 'Web 2.0' term popularized by content publisher Tim O'Reilly).

Focus: The Feature Film Screenplay

My primary content focus is the feature film screenplay, rather than screenplays intended for television (TV) or computer gaming or animation for example. It should also be noted here that any reference made to a screenplay or script generally means the "author's" or "selling" or "spec" (speculative) script and not the "shooting" script used in production, which may be substantially different in content and format than the original screenplay purchased from the screenwriter, nor any final transcript(s) produced from the released movie.

One of the earliest definitions of a screenplay (made by Barker in 1914 as quoted in Karetnikova 1990:5) is a 'Photoplay or Movie Picture play is a story told in pictures.' The *Palmer Plan Handbook* (1919:17), describes photoplay writing as a 'new art' that demands a 'new language, a new method of expressing thought and communicating emotion'. These early commentators clearly recognized the need to write visual stories to suit a visual medium. Karetnikova also highlights that the earliest scripts, such as George Méliès *A Trip to the Moon* (1902), were little more than a list of outline story headings (1990:2). The role of photoplay- or scenario-writer (now screenwriter) was already well-established in the second decade of the 20th century when Frank E. Woods, Captain Leslie T. Peacocke, Anita Loos and others were making a name for themselves and the fledgling Hollywood

movie studios were already receiving some 50,000 submissions a year according to Edwards and Skerbelis in *I Liked It, Didn't Love It* (2009: 12-15).

Today, a feature film screenplay is a structured document representing dramatic content intended for realization as an audio-visual, cinematic experience, typically of some 90-120 minutes in duration. Feature film screenplays for the cinema are different in both structure and format from screenplays intended for one-off TV dramas or a series of TV episodes, not just because they are formatted differently and may contain some different structural elements, but also because they are not subject to certain TV screenwriting conventions such as the need to write to ongoing series themes (defined by a 'bible' and overseen by the 'showrunner' role) or to structure the screenplay in a way that recognizes the key role of advertisement breaks (in commercial/cable television) as an influence on story pacing and the placement of plot points (e.g. cliff-hangers inserted to encourage viewers to return after a bathroom break or to tune in to the next episode). Movies destined for TV are also structured differently - into 7 or 8 acts according to Edwards and Skerbelis (2009:144) - as opposed to the conventional 3-5 acts for a feature film screenplay.

A screenplay is termed 'structured' because as MacDonald says in *Disentangling the Screen Idea*:

The screenplay [...] is the record of an idea for a screenwork, written in a highly stylized form. It is constrained by the rules of its form on the page, and is the subject of industrial norms and conventions. (2004b: 89).

Argentini (1998) and others (e.g. Cole & Haag (1989) and Riley (2005)), have produced style manuals and guides for defining the various screenplay elements and for formatting the script for presentation to a reader. These conventions have evolved over time and at different periods have reflected different styles of screenwriting. For example during the 1980's it was considered acceptable for screenwriters to include directions in the script (e.g. references to shots and camera angles or how the actor should deliver

their lines). Today this is considered bad form according to commentators such as Flinn (1999:109) and Epstein (2002:109-110). Screenplays not produced in accordance with the prevailing formatting conventions are likely to be viewed unfavorably by those involved in their appraisal, purchase or production since they are said to involve more effort to read or evaluate when they deviate from convention.

Screenplay content is “audio-visual” because it contains references to sound (e.g. spoken dialog and sound effects) and movement (e.g. human and non-human action) that are expressly intended for realization as an audio-visual experience for an audience. Novels contain references to sound and movement but the intention of the writer is that these references help create a picture in the mind of readers and stimulate their imagination. In a screenplay sound and movement must be translated directly into an entirely separate, tangible end product i.e. a sequence of individual film frames with an accompanying soundtrack. This change of form is a key difference between the end product of novel writing and the end product of screenwriting. The end product of a novel exists in the reader’s imagination; the end product of a screenplay is a fixed interpretation of the text (interpreted by a ‘director’) – fixed on physical film or as a digital video file format - that delivers a specific experience intended to audio-visually stimulate the audience’s imagination.

A screenplay is a relatively substantial document in terms of content. Most feature length movies are 90-120 minutes in length - seldom shorter and sometimes longer. An industry rule-of-thumb is that a page of script (in the traditional typewriting fixed-spacing font of Courier 12) roughly equates to 1 minute of produced screen time. Therefore most feature film screenplays are 90-120 A4 or US-letter size pages in length. In the past, feature film scripts, like novel manuscripts, were handwritten or typewritten. Today’s preferred fixed font for screenplays, `courier 12`, reflects this typewriter and early-computing heritage (see Millard, 2010:15-16) and the fact that a fixed font, rather than a variable-width font, is used ensures some degree of

standardization of the content so that the potential movie running-time can indicated from the number of screenplay pages.

Today, many screenwriters produce their work using a screenwriting software application on their personal computer. Dozens of screenwriting software applications are available for all the popular operating systems platforms including Microsoft Windows, Apple Mac OSX and open-source Linux. Consequently most screenplays are now stored and published as an electronic file in any of a variety of popular formats, for example: text (.txt), html (.htm), Adobe PDF (.pdf), Microsoft Word (.doc), XML (.xml) or in the proprietary format of a particular screenwriting application such as Final Draft's original file format .fdr (switched to the xml-based .fdx in version 8). Unfortunately there is no one standard screenplay file format used by all screenwriting applications. Although Final Draft's .fdx format may be considered the de-facto 'industry standard' as most other screenwriting applications can either import and/or export to this file format, due to Final Draft's perceived market penetration and industry practitioner popularity.

Datafication

Quantifying screenplay content in the form of data is one implementation of what Mayer-Schönberger and Cukier refer to in *Big Data* as 'datafication' (2013: 73-97). They make the point that, 'The act of digitization - turning analog information into computer-readable format - by itself does not datafy.' So the datafication of a screenplay does not only mean the presentation of the script in a digital format, such as in the form of an Adobe PDF file. Datafication means the decomposition of the textual screenplay content into a series of data points designed and stored to make the content machine-readable and to facilitate machine processing.

In an example relevant to screenwriting (2013: 84), the authors explain the utility of making words into data by referencing Google's effort to make as many of the world's books searchable, by 'datafying' the individual words in the texts, which enables their Ngram viewer

(<http://books.google.com/ngrams/>, accessed 26 April, 2013) to generate interesting analysis of the use of words or phrases over time.

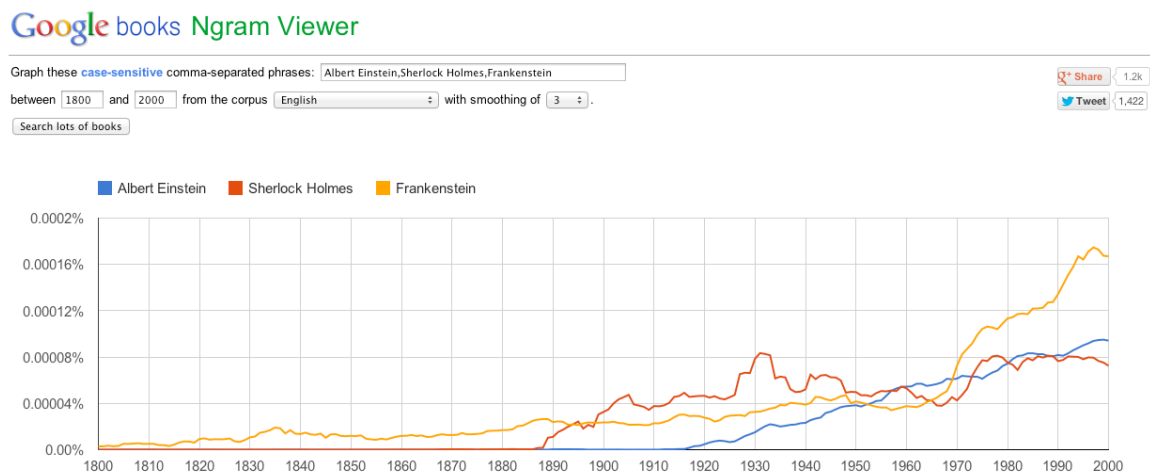


Figure 1.1 - Google Ngram Viewer plots specific name mentions over time

As more and more physical world entities and events become ‘datafied’ – locations (e.g. as latitude and longitude co-ordinates), relationships (e.g. as social network interactions), the current state of an Internet-of-Things object etc. – the datafication of screenplay content is an inevitable additional ‘brick in the wall’ and something with the potential to significantly change the practice of screenwriting and the notion of the screenplay as a digital artefact.

A key outcome of making screenplay content machine-readable is that the content is easier to analyze and visualize (terms collectively referred to as ‘analytics’ below) programmatically. By analyze I mean applying an analysis technique to the data (e.g. clustering) and by visualize I mean presenting the result(s) of that analysis in a way that facilitates better understanding of the meaning of the data (e.g. as a chart or a timeline). Screenwriting 2.0 assumes that every screenwriting application will include ‘embedded’ analytics – unlike today.

In fact, in the time it would take a typical human reader to read a single script and deliver some qualitative analysis based on experience or ‘gut-feel’, a computer could read an entire corpus of datafied scripts to process and

correlate what could be a significant volume and variety of quantifiable data to deliver insights based on the evidence of the data rather than the intuition of a human reader. Quantitative analysis becomes more important once more screenplays are datafied. Franzosi (2010:4) answers his own question ‘Words are beautiful: Why do you want to turn them into numbers?’ with this answer (2010:5), ‘I quantify simply because I have far too much information to deal with qualitatively’.

This quantitative ‘data-based’ approach does not replace the qualitative approach; it merely supplements and complements it, by adding an evidential machine-analyzed perspective to an intuitive human-analyzed perspective. This may result in a more-rounded analysis of the screenplay than either approach on its own. The potential of evidential analytics of screenplay content, enabled by screenplay as data, is a key outcome of datafication that is explored in this thesis.

Research Question

Here, the screenplay is primarily considered as a digital data artefact rather than as a digital document artefact. And also as a dynamic artefact where the data is subject to ongoing change during a lifecycle that will involve a changing ‘community of interest’ comprising various stakeholders with different content management roles.

Datafication of the screenplay has the potential to positively impact the nature of the screenplay as a digital artefact, and the way screenplay content can be analyzed and visualized and screenwriting as a collaborative, social process. Therefore the research question that informs this thesis is:

What are the possibilities of screenplay ‘datafication’?

How the screenplay as data can impact creating and managing, presenting and sharing, analyzing and visualizing screenplay content.

Creating and Managing

Creating and managing screenplays refers to how the textual content is stored and organized - for example, in a database or a file, in a script-centric paradigm or a scene-centric paradigm – and the importance of structural groupings and layers, such as sequences and storylines, for organizing and navigating the screenplay content.

Presenting and Sharing

The presentation of a screenplay should not be entirely focused on the script as a document with an emphasis on viewing the script simply as a digital version of the traditional physical printed artefact. The screenplay as online, digital artefact demands more attention is paid to the visual look and feel of the content presentation and to the variety of ways the content can be organized, navigated and searched. The social aspect of screenwriting as a collaborative, co-creation process demands that screenplays are easy to share and interact with to support the kinds of activities associated with content sharing by today's social-networking generation including commenting, rating/ranking and content linking.

Analyzing and Visualizing

The screenplay content must be stored and organized in a way so as to facilitate content analysis and content visualization in order to support deriving 'insight' from any quantitative analysis of the screenplay data. The availability of content analysis and visualization should be a key tool in helping the screenwriter to 'ask more questions' of their script in order to improve its quality and /or commercial value through analysis of content from different perspectives.

To investigate this question, my research has focused on three areas each representing a chapter in this thesis:

1. Approaching the screenplay as a data, not document artefact or 'screenplay as data'.

2. Recognizing that screenwriting is a social and business process rather than an individual and solely creative process or 'screenwriting as social network'.

3. Leveraging the availability of the screenplay as data for analytical and visualization purposes or to deliver 'screenplay analytics'.

My final chapter, before those discussing the practical deliverables from my research, is 'screenplay as design' where I discuss how a datafied screenplay provides the platform that facilitates possible ways to design and automatically generate the 'framework' for a screenplay.

Screenwriting 2.0

When I use the term 'Screenwriting 2.0' I am proposing a term that refers to an evolutionary development in screenwriting technology and practice analogous to that characterized by the term 'Web 2.0'. The Web 2.0 term has of course been hijacked by the technology press as a marketing term after it was coined by Darcy DiNucci in *Fragmented Future* (see http://www.darcy.com/fragmented_future.pdf, accessed 17 Dec. 2013) and popularized by Tim O'Reilly's Web 2.0 conferences from 2004 onwards (see *What Is Web 2.0?* <http://oreilly.com/web2/archive/what-is-web-2.0.html>, accessed 17 Dec. 2013). Web 2.0 has come to mean the convergence of a number of technologies – including, among others: the cloud, mobile devices, global positioning satellites (GPS), pervasive social networks, improved data visualization, a focus on website design and usability - and ways of using the web that denote a meaningful shift away from the earlier versions of the Web.

Back in 1999, DiNucci suggested that 'Today's Web is essentially a prototype, a proof of concept.' (1999: 32). I suggest that the same could have been said about screenwriting applications in 2006 - applications that could generally be characterized as allowing a single user to access the application to write a screenplay document, formatted to industry conventions, and to

print it out on paper or share electronically in the form of a PDF file. As Andrew Kenneth Gay says, 'Given access to a time machine, a Hollywood screenwriter working half a century ago would encounter little difficulty adjusting to the conventions of professional screenwriting in 2012' (see *Screenwriting 2.0 in the Classroom? Teaching the Digital Screenplay*, <http://framescinemajournal.com/article/screenwriting-2-0-in-the-classroom/>, accessed 17 Dec. 2013) to which we might add adjusting to the technology of screenwriting.

When I began my research in 2006, few screenwriting applications stored their screenplay content as data (in a database as opposed to in a file), only one (to my knowledge) offered any kind of screenplay analytics, and none were designed to operate as a multi-user social network. To me, these are at least three hallmarks of the evolution towards Screenwriting 2.0 that differentiate this iteration from the first generation of screenwriting applications or Screenwriting 1.0 if you will. The fundamental purpose of this thesis is to discuss these hallmark characteristics and apply them to the creation of a Screenwriting 2.0 prototype application.

Sources

I have selected four information sources for my literature review, which of necessity encompasses non-literary artefacts such as software tools and applications:

1. Academic research focused on, or applicable to, screenplay content management and the technology of screenwriting.
2. Screenwriting, novel- and play-writing "how-to" manuals and other works relevant to screenwriting, which are not considered 'academic' but are primarily intended to help writers produce successful, that is commercially valuable, screenplays.
3. The current generation of story-planning and scriptwriting software applications and tools available in the marketplace.

4. A range of generic textual analysis and data visualization tools that are not specifically designed for analyzing or visualizing screenplays but have the potential to be applied to screenplays.

Most of the non-academic literature relating to screenwriting is primarily concerned with how to write a "better" screenplay and generally "better" is taken mean a screenplay that has an improved chance of succeeding commercially. This is understandable, because unlike a novel, screenplays are written to make into movies rather than read for pleasure. So unless a screenplay is bought it is unlikely to be produced, which usually defeats the purpose of writing it. However, this commercial bent does not mean that how-to manuals have no merit.

But what should be noted about many screenwriting-related how-to manuals is that any screenplay analysis they contain is often not of a script text per se – although this is not the case for manuals focused on specific script-improvement tasks, such as writing better dialog for example - but of the movie that was made from a script: Movie content analysis rather than screenplay content analysis. This is an important distinction because the focus of this analysis is based on the visual end product rather than the textual screenplay (i.e. the output rather than the input). It could also be argued that analyzing a visual movie is easier than analyzing the script itself because whereas almost anyone who watches a movie can describe the plot(s) and characters of a movie just from seeing it, it is harder to do this by only reading the script without having seen a movie produced from it.

The realized visualization of the movie itself assists the viewer to raise the level and quality of their analysis of its content. Just as handling or using a finished product aids the quality of analysis as opposed to simply reading the product's technical specification or viewing a mockup sketch or photograph of the product. To analyze screenplay textual content is simply not the same thing as analyzing audio-visual movie content derived from the screenplay. They are two different analytical subjects.

Methodology

This is a PhD by practice that is framed as a software development project. The deliverable of the 'practice' element is a series of prototype applications that relate to the research question. These prototypes are informed by the research into screenwriting theory and practice as represented by the 'written' part of the thesis. The prototypes are working applications that were/are delivered online and publicly accessible to anyone who registers to use them.

Start Point

Inevitably my thesis is informed by my own 'start point', which when I started this PhD in 2006 included a MA in Screenwriting, a MSc in Organization Consulting and my 'day job' as an enterprise application business analyst.

My MA course provided me with a basic understanding of screenwriting practice. But what I noticed was that the content of the course lacked any focus on the technology of screenwriting. And apart from an assignment to write a 'script report', assuming the role of a script reader, there was no other discussion of screenplay analysis let alone the use of technology to deliver it. This all struck me as odd, given that we live in a digital age, many people on the course wrote their screenplays using screenwriting applications and the analysis of many other kinds of business data was, and continues to be, a focus in the continuous improvement of business software applications generally.

My MSc was focused on the relational and reflective aspects of consulting. We were encouraged to view consulting as collaboration – between client and consultant – in which any outcomes, whether positive or negative, are co-created. There was an emphasis on the dynamics of consulting and paying close attention, both 'in the moment' and 'on reflection' to what you and the client were actually saying and doing. One of the methodologies we

experimented with was Action Research, which I will return to later.

My daily work as a business analysis often involves me in projects to select and/or implement relatively complex enterprise resource planning (ERP) software. A core focus of these kinds of projects is understanding the scope of operational domains in a business, the data that is managed and the processes used to create and manage that data, including user roles and activities. This perspective also informs the technology deliverables from my thesis.

The lack of technology emphasis on my MA course led me to investigate what technology was available for screenwriters. Some formal output from this investigation was a series of articles I wrote for *Scriptwriter* magazine in 2006 that reviewed specific screenwriting applications and the screenwriting technology market generally (see McKie 2006a and 2006b for two examples).

From my investigation and trial use of a number of commercial screenwriting applications it became clear that the technology was limited in scope and that among the functional deficiencies were the lack of content analytics and the management of screenplays as a single user 'closed document' artefact rather than a social 'open data' artefact. I was particularly interested in the possibilities for screenwriters to gain more insight about their scripts through the use of content analytics, something that only one screenwriting application – *Sophocles* - really addressed in any significant way at the time.

Analyzing the content of screenplays, by leveraging the wide range analytic and visualization technology available for data analysis, was not easy simply because although many screenplays were available in a 'digitized' form (i.e. available in an electronic file format such as Adobe PDF) they were not 'datafied' (i.e. available as data in a database). This meant that in order to convert a digitized script into a datafied script generally required some manual 'wrangling' or programmatic intervention to get the data into the

format needed so that analytic/visualization tools could be used to analyze the data content of the scripts. Clearly a significant issue was that the prevailing focus in screenwriting applications on 'screenplay as document', rather than 'screenplay as data', was holding back the development of screenplay analytics (the analysis and visualization of screenplay content).

It was clear from my initial research into the 'manual' (rather than automatically generated by software) visualization of screenplay content in various academic and how-to manuals that there were examples of visualizations, particularly of the analysis of screenplay structures, that could be possible outputs generated from screenplay analytic functionality in screenwriting applications. Samples of these visualizations are discussed in chapter 2.

From discussions in 2006-7 with my then supervisors Prof. Fionn Murtagh and Adam Ganz, screenplay analytics seemed like a useful line for further research, and in fact this early focus did lead to a number of tangible outputs from both my supervisors and myself.

Murtagh went on to produce a number of papers that focused on the content analysis of both movie and TV series screenplays (most of these papers are listed here - <http://www.narrativization.com> - accessed June 17, 2014).

Ganz organized *Film, Visualization, Narrative*, a workshop run at Royal Holloway, University of London on 17 Nov. 2006, supported by the AHRC ICT Methods Network and LCACE (London Centre for Arts and Cultural Enterprise) and then edited the *Reconstruction* Vol. 8, No. 3, 2008 issue focused on Visualization and Narrative, which included my own contribution *Screenplay Visualization: Concepts and Practice* (see <http://reconstruction.eserver.org/Issues/083/mckie.shtml> - accessed June 17, 2014).

I presented Scriptcloud - my first screenplay analytics prototype application - at the Second International Workshop on Semantic Media Adaptation and Personalization (SMAP) held at Brunel University in 2007. My paper,

Scriptclud (sic).com: Content Clouds for Screenplays was also included in the conference proceedings (see <http://dl.acm.org/citation.cfm?id=1339060> - accessed June 17, 2014). I also secured LCACE funding for Scriptgeist, my next screenplay analytics prototype application released in 2008, with the help of Lydia Daniels of the Royal Holloway Enterprise center and supported by Adam Ganz.

This early work on screenplay analytics was also referenced in a Nature online news item *Here's looking at you, kid. Software promises to identify blockbuster scripts*, (Nature, vol. 453, p. 708, 4 June 2008: see <http://www.nature.com/news/2008/080604/full/453708a.html> - accessed June 17, 2014).

Next Steps

At this point, my research had what appeared to be a useful focus – screenplay analytics – so there were a number of ways I could have proceeded in terms of my research methodology in the context of my framing of my PhD by practice as a software development project.

One might have been a case study approach, whereby I studied a specific screenwriters practice or a small group of screenwriters practice in order to extrapolate ‘learnings’ from these case studies to inform my software development. Although I solicited potential participants to work this way via the Internet no-one was really interested enough in screenwriting technology or screenplay analytics to devote the time and energy needed to participate in this way and as I did not know any suitable participants personally, I had to reject this as a way forward.

I could have informed my development by use of one or more surveys about peoples’ views on the technical development of screenwriting technology. In fact I did a basic online survey on this topic just before the start of my PhD, the results of which were summarized by screenwriting blogger Alex Epstein here:

<http://complicationsensue.blogspot.co.uk/2006/11/polling-numbers.html>, accessed 14 March 2014.

Unfortunately, in general in the Internet age, many people are 'surveyed out' due to the many survey solicitations they now receive by email (I myself get at least one a week), and screenwriters from active online communities like *Shootingpeople.org* did not seem that interested in responding to surveys and particularly not when the subject-matter was technology-centric. The survey referenced above attracted less than 50 respondents. Surveys are also difficult to design effectively and interpret usefully if you are not a professional statistician. So again I did not feel this was a direction worth pursuing.

I could also have recruited some kind of 'beta test' group from the start and tried to involve them throughout my research. But this was sure to be a big ask given that my part-time studies could take 6-7 years and few people will remain engaged with someone else's project for that period of time unless it has a significant value proposition for them. In fact towards the end of my studies I managed to recruit a small beta test group of interested academics/students to test a Scenepad prototype deliverable, but only one of them truly engaged with the project over the beta period, which in this case was only 3 months and participants were hand-held along the way.

So I decided to start like so many software developers of online applications start – by building a prototype, making it available online and seeing if anyone would use it. The hope was that if I built it 'they would come'. If I was lucky, maybe it would 'go viral' in some way, as certain applications that somehow appeal to the zeitgeist manage to do every now and then. This is a risky approach as, without any marketing funding to promote the applications, usage depends largely on positive 'word of mouth' marketing from users themselves. In essence the idea was to let the application prototypes 'speak for themselves' and see what emerged as a result. So this became my start point.

What must be made clear here is that I did not expect anyone to engage with my prototypes. In a sense the development and delivery of the online prototypes is best characterized as a kind of 'thought experiment'. If I was lucky, some people may engage with the prototypes and provide useful feedback for further development but if not this would not invalidate the purpose of the exercise. Without anyone using the prototypes I may not be able to observe actual use of the software or conduct genuine experiments with the functionality delivered (e.g. A/B testing commonly used for validating website page design – see for example <https://www.optimizely.com>, accessed July 09, 2014) but the prototypes do function as a series of 'imaginary' scenarios in order to imagine what a datafied screenplay could deliver in terms of analytic and social potential.

Qualitative Research Focus

The focus of my research is more qualitative than quantitative. For example, it was not about gaining as many users as possible for my prototypes, or posing user surveys to garner 'votes' for features and functions, or gathering statistics about the usage of specific features and functions in my prototypes, simply because without statistically significant user engagement with my prototypes this kind of quantitative analysis is impossible.

It was however, about the kind of qualitative research as defined by Savin-Baden and Howell Major (2013:11): 'We define qualitative research simply as social research that is aimed at investigating the way in which people make sense of their ideas and experiences.'

In my research context I correlate *ideas* with 'screenplays' and *experiences* as the experience of 'developing and writing a screenplay'; the tools to help *people make sense* comprise the software prototype deliverables themselves and the users' use of them.

My research approach was informed by the principles of 'action research', originally attributed to Lewin (1946:35), which I had been introduced to and

used during my MSc in Organization Consulting at Ashridge. In terms of a specific definition of action research, I refer to that of Kemmis (2012:417) as quoted in Savin-Baden and Howell Major (2013:245): 'Action research concerns action, and transforming people's practices (as well as their understandings of their practices and the conditions under which they practise(*sic*))'.

In my research context I correlate *action* with 'delivering software prototypes for use' so that the functionality that is delivered has the potential to be used for *transforming people's practices*.

Action research implies the use of a spiral cycle of activities that broadly speaking comprises planning things (plan), doing things (do or act), observing the results (observe) and reflecting on the results (reflect) in order to (in some cases) reiterate the cycle one or more times to achieve a qualitative improvement. Again, in my research context, I am more aligned with a variant of the approach outlined by Jean McNiff (see <http://www.jeanmcniff.com/ar-booklet.asp>, accessed 11 March 2014):

- identify an area of practice to be investigated i.e. screenwriting
- imagine a solution i.e. new screenwriting technology
- implement the solution i.e. a screenwriting application prototype
- evaluate the solution i.e. in the light of user's feedback/usage
- change practice in light of the evaluation i.e. refine existing or deliver a new prototype

An issue with the use of action research is whether the intention is to improve the practice of the action researcher or that of the group of practitioners that the action researcher involves in the action research project. Since in this case it was in a sense both, then the latter intention may surface a problem identified by Savin-Baden and Howell Major (2013:254):

In some forms of action research, where the researchers design the action plan for the practitioners (rather than with them), there is a danger that there will be little ownership by the practitioners and this in turn will affect the degree to which change is 'owned', implemented and really possible.

This is a potential problem in any 'speculative' software application development where the software deliverable, i.e. the *action plan* as prototype, is designed by the developer and presented 'as-is' to potential users. It was a problem I faced but unfortunately the general lack of any user feedback meant that it was not an easy problem to overcome since so few users engaged with the prototypes themselves or with me as the developer - despite being provided with online feedback forms and my offline contact numbers and email address.

Data Collection and Documents

In my research context, data collection was largely about understanding:

1. how many people were using and engaging with my prototypes
2. what kind of people were signing up to use the prototypes
3. what they liked and disliked about the features and functionality
4. what ideas and suggestions they had for changes or new functionality
5. how many scripts they uploaded into and/or wrote in the applications

I could determine (1) from how many users had signed up and how often they logged into a prototype since this data is collected by the system itself. But in practice many registrants did not provide much in the way of metadata about themselves in their user profiles so (2) was unreliable. And although it is easy to log how many page views a specific page gets in an online application, this does not really help with understanding (3) in a qualitative way. Both (3) and (4) depended on the receipt of verbal or virtual feedback from users via contact forms or feedback forms or comments within the application.

(5) is an important metric but one that essentially turned out to be pretty much a one-to-one relationship (i.e. 1 user, 1 script). Professional screenwriters who might be fortunate enough to be managing multiple scripts at a time were unlikely to risk using these unproven tools and most people who actually tried the applications probably only had one screenplay that they were currently focused on and prepared to 'dabble with' in the applications. So this data point also proved to deliver minimal insight.

Fieldwork, Interviews and Observations

Despite the deliverables being online applications used by remote users (albeit from all over the world) and therefore functioning as 'surrogate' fieldwork, I did some activities that could be classed as traditional 'fieldwork, interviews and observation' as part of my research. This fieldwork took place in the context of working with three small groups of students from the 'teaching and learning' community of interest, as outlined below. In all three cases I worked face-to-face with the students asking questions, observing behavior and discussing any comments posted directly to the system by the students or given verbally.

My interviews and observation, a mixture of 'focus-group' (or *constituted group* – see Bruhn Jensen 2002:241) and 'one-to-one' oriented, did not follow a script or present a structured checklist of questions for students to answer or formally record peoples' behavior in any ethnographic fashion but were more informal (although all three required significant preparation work to 'design' the formal part of the sessions):

- At my University College Falmouth session (c. 2 hours), I presented some principles of screenplay analytics, gave a demonstration of my Scriptgeist prototype and then discussed the utility of screenplay analytics with the students in a Q&A session.
- At my Bournemouth University session (c. 2 hours), I gave a demo of my Scenewrite prototype and then facilitated the students use of the system in a 'floorwalker' training role so they could engage with the system immediately. One student did continue to engage with the Scenewrite prototype post-session but subsequently gave up and did not answer emails.
- At my Royal Holloway session (c. 3 hours), I introduced and demonstrated my SceneClass add-in for Wordpress, facilitated the students' use of the system in a 'floorwalker' training role and then reviewed their posted comments and experience of using the system in the closing feedback session.

In terms of Savin-Baden and Howell Major's continuum of observation roles (2013:394 figure 25.1) I would say my personal role in these sessions was located between 'balanced participation' and 'active participation'. It could

not involve 'complete participation' since the point was for the students to respond to/use the system prototypes, and not me. However, I think it is fair to say that fieldwork, interviews and observation did comprise a minor but demonstrable activity within my research methodology.

Data Analysis and Interpretation

The data collected from my prototypes during my research was hardly significant enough to draw any meaningful conclusions from other than 'anecdotal' observations such as:

- User engagement went down as functionality went up
- Users showed more engagement with a screenplay oriented online game (*Geistmeister* in *ScriptGeist*) than with actually using prototypes to write and analyze screenplays
- Screenwriters do not seem to value the utility of analytics to help them 'ask more questions of their scripts'.

Whereas my first prototype, the word frequency cloud generator Scriptcloud, attracted a few hundred users, my final prototype 'submission' deliverable Scenepad – containing significantly more functionality including screenplay writing capability - attracted only a handful and an earlier iteration of Scenepad did not manage to sustain the engagement of a small group of motivated beta-test users who had signed up to participate in a highly structured beta test program.

My Geistmeister screenplay guessing game that was part of the Scriptgeist prototype generated much more engagement than the rest of the application that actually delivered significant (in terms of comparable commercial applications) screenplay analytics functionality.

Anecdotal responses to my positing of the potential usefulness of screenplay analytics on blogs and forums were not positive. Of the very few blogger comments I received, at least two specifically questioned the need for 'innovative' screenwriting technology or the utility of screenplay analytics as

shown below in the screenshots below:

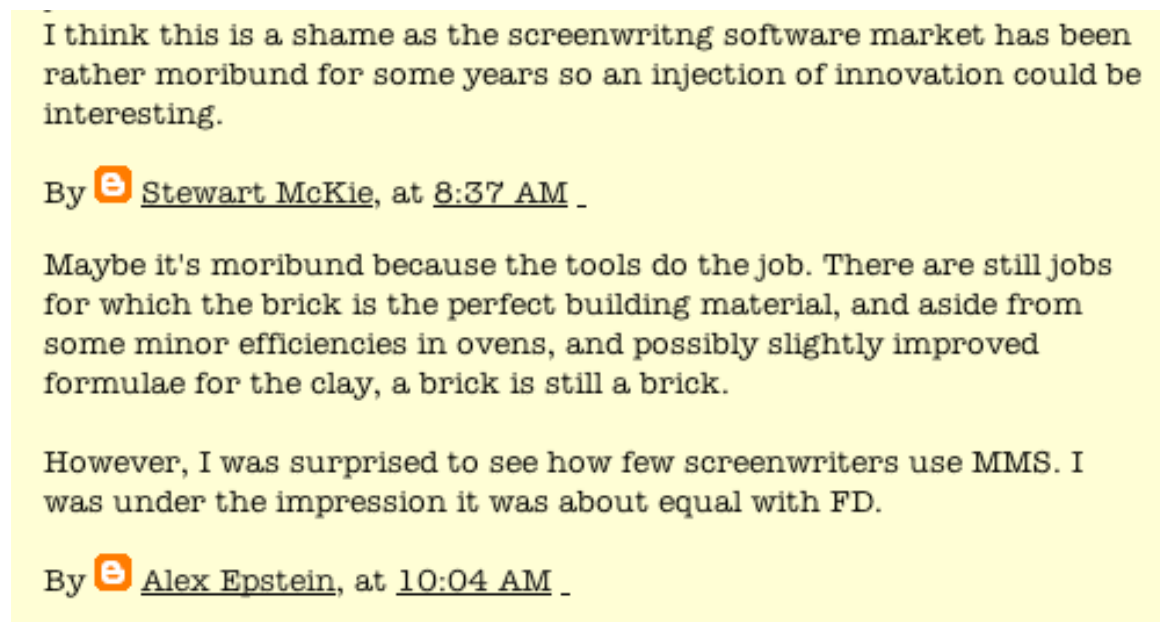


Figure 1.2 – Comment posted by Alex Epstein on his *Complications Ensue* blog

(see <http://complicationsensue.blogspot.co.uk/2006/11/polling-numbers.html>, accessed 14 March 2014.)

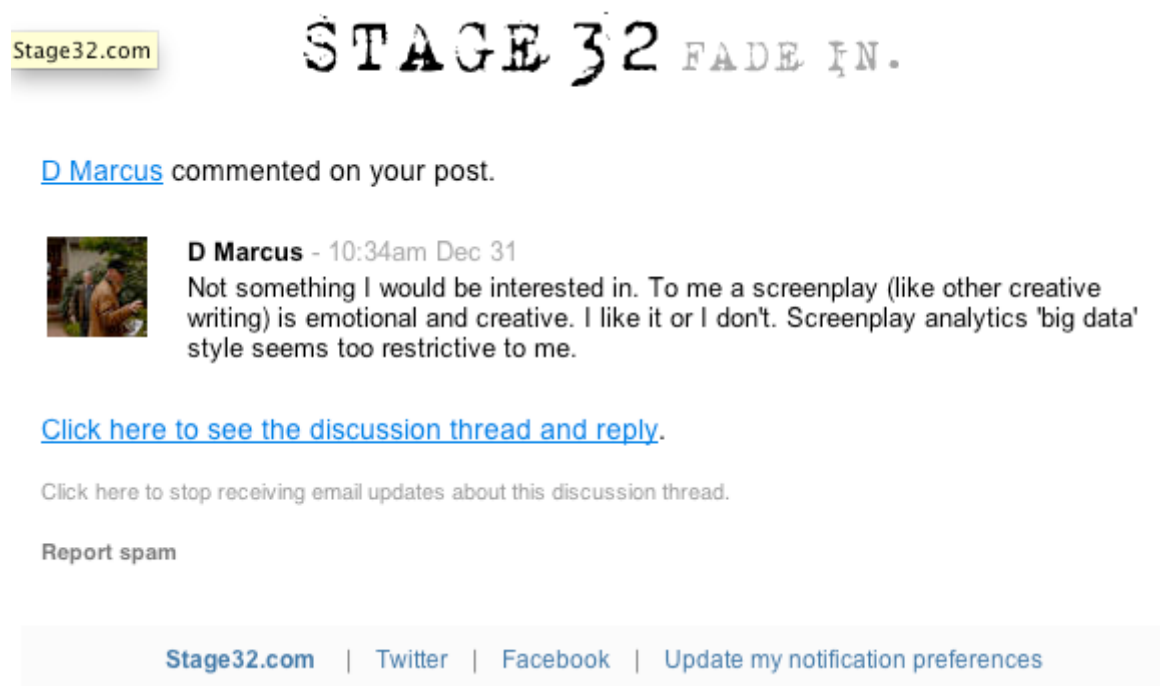


Figure 1.3 – Comment posted by a Stage32.com user on their blog and sent to me by email on 31 December 2013

Since there was so little data collected, either from ‘external’ blog comments or as ‘internal’ feedback on the applications themselves, there was no point in to trying to draw any conclusions from it, so data analysis and interpretation turned out to be minimal.

Prototype Development Methodology

The development of the software prototypes in response to my research question was a key deliverable of the practice element of this PhD. In most commercial/business settings, a software application development project is generally one of two kinds:

1. An application development carried out as a speculative venture by a software developer/development firm where the potential user community is usually not specifically known.
2. An application development that is internal to a specific organization carried out by internal/external resources and where the actual user community is specifically known.

In the case of (1) the development project typically involves one or more ‘developer’ roles – coders, user interface designers etc. – but may not involve any business analysts or actual users other than possibly a few ‘beta’ users who have agreed to participate in some kind of testing program prior to the application’s formal release.

In the case of (2) the development project typically involves three parties:

1. The developer(s)
2. The business analyst(s)
3. The business user(s)/subject matter or domain expert(s)

Case (2) may be a more complex and ‘overseen’ project because often more people/roles are involved and specific departmental/organizational budgets are being consumed to fund the work. Although case (1) projects can be equally large and complex, today many so-called ‘app’ developments can

essentially only involve one person, which is the case here.

So here, unlike many case (1) or (2) projects, a team of developers did not create the deliverables. In most cases, my prototype deliverables involved a partnership – involving myself as ‘designer’ and ‘data modeler’, working with a developer partner as ‘coder’. The users were represented only by those people who managed to find and sign up for the applications online. In this specific case, I represented the developer, the business analyst and the (screenwriting) domain expert. In reality this is not necessarily a good idea as it tends to encourage solipsistic practice.

As with virtually all software developments, large chunks of code were not developed either by myself or by my coding partners but by others who have made their efforts available to all through various kinds of open source and commercial licenses. Use of or adaptation of these open source/licensed components is standard practice in the commercial software industry.

There are essentially three skillsets used to deliver the kind of software application prototypes necessary for this PhD, reflecting the so-called ‘3-tier’ development architecture comprising:

1. Presentation tier skillset: user interface design and coding
2. Business logic tier skillset: business rules /event design and coding
3. Data management tier skillset: database design and coding

In terms of the early prototype deliverables I acted as the ‘designer’ for all three tiers and the ‘coder’ for tier 3. As I became a reasonably proficient PHP programmer myself during my PhD studies, I acted as the designer and coder for all three tiers in the later deliverables.

Because a development team was not involved in the creation of the prototype applications there was no need to use one of the formal software application development methodologies that are usually adopted in team-centric developments, for example the Scrum methodology (see Sims 2012). However in order to relate some of what was done to a popular software development methodologies, I will indicate how there was some overlap

between my practice and that of certain software development methodologies.

In Scrum role terms I acted as a combination of 'product owner' and 'team member'. However since no formal team-centric Scrum process was followed I did not perform the role of 'scrum master' since no actual scrums took place. No Scrum artefacts, such as user stories, backlogs and tasks were created nor was a scrum 'sprint' cycle followed.

In terms of the Agile manifesto (see <http://agilemanifesto.org/principles.html> accessed 5 February 2014) only two of the 12 principles are applicable here:

Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.

Working software is the primary measure of progress.

And in terms of Agile values, both 'Working software over comprehensive documentation' and 'Responding to change over following a plan' informed the deliverable developments. Similarly, in terms of the *Crystal Clear* properties (Cockburn 2005:17), my practice espoused both 'Frequent Delivery' and 'Reflective Improvement'.

But fundamentally the core methodology was simply that of an iterative development throughout the duration of my research, whereby each prototype iteration became functionally richer as my research progressed with the final 'submission' deliverable being the most functionally rich of all. None of the functional development was in direct response to user request/demand since this did not happen.

Development Choices

As with any software development it was necessary to make choices as to the technology used to develop with and the platforms targeted to develop for. Given that this was largely a development managed and delivered by

myself, it is natural that my own skillset and experience played a significant role in my development choices.

From a practice perspective, I faced some challenges. Any screenplay content needed to be 'datafied' and this datafied content had to be accessible to analytic and visualization tools. I had to be confident that I could manage this datafication and that the datafied content itself would be reasonably easy to analyze using off-the-shelf tools. I am not a computer scientist so I am not trying to invent new algorithms for these purposes, merely to use readily available, popular and low-cost or free open source technology wherever possible.

In terms of the 3-tier application delivery architecture outlined above there are at least three main development technology choices to be made:

1. The delivery user interface (UI) and target user device
2. The coding language to be used for the client and server business logic code
3. The database management system (DBMS) to be used for data storage and management

At the time of my PhD, the most popular and pervasive, delivery user interfaces and devices include:

1. Web UI – including any device capable of running a web browser
2. Mac – the Apple Mac operating system (OSX) and Mac computer
3. Windows – the Microsoft Windows operating system and a PC
4. Linux – the Linux operating system and any device capable of running Linux
5. IOS – the operating system for the Apple iPhone and iPad devices
6. Android – devices that support Google Android operating system

Developing for (5) and (6) requires a relatively specialist and expensive skillset that I did not have and also is intended to create applications primarily designed to operate on mobile phones and tablets. There are screenwriting applications that run on mobile phones and tablets but in my view this is not the primary target device for this kind of application use case.

Developing for 2-4 also requires a relatively specialist and expensive skillset that I did not have but clearly these are key target devices for screenwriting applications as many commercially available screenwriting applications were designed to run on Macs or Windows PCs or both.

I decided to focus on (1) because I had prior experience of developing web-based applications, web applications run on most devices (both 'tethered' and mobile) and the costs and barriers to entry are low.

There are a number of programming languages I could have used for client and server side development but I decided to use JavaScript/HTML/CSS on the client and PHP on the server as again I had some familiarity with these languages, they are free to use and there are many free/low cost components available in the open source community to facilitate development using these languages. I certainly did not want to have to learn a wholly new set of programming skills as part of my PhD by practice.

There are a number of databases (e.g. SQL and NoSQL) or file systems (e.g. XML) that I could have used for data storage and management but I decided to use a structured query language (SQL) based relational database management system (RDBMS) as the database engine because I understand SQL and how to design and work with relational data models. I chose to use the open source MySQL database for data storage because it is reliable, scalable and free to use and also available on the Internet Service Provider (ISP) hosting account that I use to host and deliver my online applications.

Community of Interest

Any software development essentially involves two parties: the developers/analysts and the users. In the development of software within a specific business these parties are easy to define – usually the developers/analysts sit within the IT department and the users sit within the operational line of business units. In this case, my partner coder and I functioned as the developer party but a decision needed to be made as to

who to target as the user party.

Here my methodology is informed by Etienne Wenger's concept of a community of practice (see <http://wenger-trayner.com>, *A brief introduction to Communities of practice*, accessed 5 February 2014). My 'domain of interest' is that of screenwriting in which the 'shared competence', in relation to my research question, is specifically that of writing and/or analyzing movie screenplays using a software application. Clearly there are at least four potential user communities that exist in this domain with specific kinds of competency focus:

1. Writers – i.e. practicing amateur or professional screenwriters who write or want to write screenplays for a hobby or a living.
2. Analysts – i.e. individuals in the business of analyzing screenplays including professional agents, readers and other 'story analysts'.
3. Teachers & Learners – i.e. those involved in the teaching and learning of screenwriting practice at school/college/university.
4. Producers – i.e. the cast and crew engaged in realizing an audio/visual movie product from a script that acts as the production 'plan'.

I initially rejected (4) as a potential user community. The reason for this was that the 'producing' community is less interested in writing and analyzing the content of a screenplay and more in the practical effort of converting the screenplay into a finished movie. In software functionality terms, this user community is more concerned with tasks such as generating prop lists and daily call sheets and with managing script revisions and monitoring the production budget. This was not functionality that I initially intended to deliver and so this community was not a good fit in terms of my research question.

I initially thought that (1) and especially (2) would be my primary focus. However, the anecdotal response to contacts I made with a handful of practitioners I knew or reached out to in the 'analysts' domain was not positive. The general view seemed to be a lack of interest in tools that helped to analyze a screenplay, whether because this was seen as

antithetical to the author/analyst's 'creative' process or some kind of threat to the validity or commercial viability of fee-based qualitative screenplay analysis services.

This feedback led me to focus initially on (1) and as I did not know many screenwriting practitioners personally I focused my efforts on recruiting a user community by reaching out to various screenwriting-focused bloggers and those online communities that appeared to include a significant number of screenwriters. Usually this was by means of a direct contact email, posting comments on other people's blogs or a posting a solicitation to try out a prototype on the online forum of a screenwriting user community.

One screenwriting blogger who did engage with me initially was Alex Epstein, author of the *Crafty Screenwriting* books. He made a few posts that referred to my work that attracted a handful of comments from members of his blog community– see:

- <http://complicationsensue.blogspot.co.uk/2006/11/polling-numbers.html>
- <http://complicationsensue.blogspot.co.uk/2007/03/ive-looked-at-clouds-that-way.html>
- <http://complicationsensue.blogspot.co.uk/2008/08/scriptgeist.html>

(All above accessed 5 February 2014)

I also joined and participated in the online community at <http://shootingpeople.org> to solicit screenwriters to try the initial scriptcloud.com and scriptgeist.com prototypes. As this community requires users to pay a subscription fee to participate, I assumed this would be indicative of higher quality and more engaged participants. And in fact many of the hundreds of users who signed up to use my early prototypes did originate from the *Shooting People* community. However, it appeared that over time people became disinterested in the potential of screenwriting 2.0 technology. In addition, no blog posts I made on my own blog at phd.tripos.biz or any of my prototype applications ever gained much feedback from users and certainly never went 'viral' in any way.

What quickly became clear to me was that any user community I could engage with was less a community of practice and more a community of interest. As Wenger states (p.2), 'A website in itself is not a community of practice' and 'A community of practice is not merely a community of interest--people who like certain kinds of movies, for instance.'

So despite providing feedback forms on my prototypes for users and open commenting on my blog (originally) I quickly found that the few users who did engage with the prototypes were not that interested in developing *shared practice* in the context of my research question but merely in 'dabbling' with it. It was clear that I was working with a kind of community of interest (CoI), along the lines of that discussed by Fischer, (see Gerhard Fischer, *EXTERNAL AND SHAREABLE ARTIFACTS AS OPPORTUNITIES FOR SOCIAL CREATIVITY IN COMMUNITIES OF INTEREST* <http://l3d.cs.colorado.edu/~gerhard/papers/ccmcd2001.pdf>, accessed 5 February 2014) rather than a community of practice (CoP).

Fischer (p.4) states that communities of interest (CoIs):

are "defined" by their shared interest in the framing and resolution of a design problem. CoIs often are more temporary than CoPs: they come together in the context of a specific project and dissolve after the project has ended. CoIs have great potential to be more innovative and more transforming than a single CoP if they can exploit the "symmetry of ignorance" as a opportunity for social creativity.

I had hoped that the apparent 'symmetry of ignorance' about, in particular, social screenwriting and screenplay analytics (via software) might lead to some social creativity. But I was proved wrong.

Certainly my potential CoI did not satisfy Fischer's assertion that 'CoIs [Fischer, 2001] bring together stakeholders from different CoPs to solve a particular (design) problem of common concern', since it is doubtful whether either the 'developers' or 'users' in this case can be considered to be examples of CoPs. And perhaps this was because I did not recognize or confront a specific difficulty faced by a CoI, according to Fischer:

Fundamental challenges facing CoIs are found in building a shared understanding [Resnick et al., 1991] of the task-at-hand, which often does not exist at the beginning, but is evolved incrementally and collaboratively and emerges in people's minds and in external artifacts.

The concept of Screenwriting 2.0 and specifically the practice of screenplay analytics on datafied scripts were not mainstream ideas when I started this PhD and I largely failed to facilitate the incremental and collaborative evolution of either with those users that did engage with my prototypes.

Remote vs. Local Usage

Another issue that every developer of speculative online applications faces is that their user community is by definition remote and often, as in this case, not linked by any shared bonds e.g. part of a family or business or other recognizable offline community. This led me to reconsider reaching out to the 'teachers & learners' user group since in this case I expected to be able to work locally and face-to-face with small groups of users, rather than only remotely. As a result, I engaged with 3 specific 'educational' user groups:

1. An after-school screenwriting class at my local secondary school (Shaftesbury School in Dorset) comprising year 9-11 students.
2. Undergraduate-level student groups at two universities (Bournemouth and University College, Falmouth).
3. A graduate-level student group at Royal Holloway College, London.

In the case of (1) and (2) the groups were shown the Scenewrite prototypes, and one or two students within each group subsequently used the applications to write and analyze their own screenplay scenes. However no student continued to use Scenewrite after an initial try.

In the case of (3), all students did actively engage with the SceneClass prototype (a Wordpress add-in) during the session. Yet despite clearly favourable feedback on the day, again no student continued to use SceneClass after their initial try. On the basis of this experience I decided not to focus more attention on the potential teaching and learning user community as it was not clear that the results with these three user

communities face-to-face were likely to be significantly more positive than working with a remote user community.

It is fair to say that despite numerous remote and local attempts to create and evolve a community of practice or a community of interest, I was unsuccessful at doing this with any of the three possible 'target' user communities I identified at the start of this PhD. In fact it seemed to me that the more functionality I delivered - that in my view, validated the utility of screenwriting 2.0 and screenplay analytics in particular - the less users engaged with the prototypes and the fewer users that were attracted to even try the deliverables.

Prototyping as a Development Methodology

There are a number of software development life cycle (SDLC) models that can be used to inform software application development including Agile, Big-Bang, Iterative, Prototyping, RAD, Spiral and V-Model. My intention is not to discuss these models here, as they all have pros and cons in relation to a specific development project, but simply to state that I chose to use Prototyping as my primary informing model.

In a presentation about chapter 8, *Prototyping and Rapid Application Development*, from the fifth edition of their book *Systems Analysis and Design*, Kendall and Kendall (2002) define prototyping as 'an information gathering technique and as useful in seeking user reactions, suggestions, innovations and revision plans' (see slide 8-3 in <http://www.mgt.ncu.edu.tw/~ylchen/sasd-slide/chap08.ppt>, accessed 11 March 2014).

Like Kendall and Kendall, Beaudouin-Lafon and Mackay in chapter 52 of *Prototyping Tools and Techniques* (see <https://www.lri.fr/~mackay/pdffiles/Prototype.chapter.pdf>, accessed 11 March 2014) also emphasize (p.1-2) the utility of a prototype to support 'creativity, communication' and 'early evaluation' (of design options). They differentiate between 'offline' and 'online' prototype representations that

either require or do not require a computer. They also mention 'evolutionary prototypes' (p.1-6) i.e. prototypes that evolve into a final application deliverable and sometimes involve 'participatory (also called Cooperative) design' (p1-7) whereby users collaborate with designers to more or less co-create the application together.

In *The qualitative research process*, Bruhn Jensen (2002:236) makes the point that 'Qualitative researchers tend to conceive of their studies, most generally, as an iterative or repeated process', which fits well with the practice of evolutionary prototyping in software development. And in Bruhn Jensen's terms, prototyping can be seen as performing both a 'sampling' function in the sense that each prototype presents a sample of functionality that users may respond to or not and each prototype also creates a 'naturalistic context' since by definition the application itself is the context in which the users work.

My prototypes are functional, interactive online applications, not offline, static and non-functional 'wireframes' or paper-based diagrams merely indicative or illustrative of a user interface or functionality. And my prototypes are evolutionary in nature because, iteratively, they became more functional over time - from Scriptcloud to Scenepad (as discussed in more detail in chapter 5 below - *The Road to Scenepad*).

The table below shows how the application functionality was evolved from basic word clouds in Scriptcloud, to analytic charts in Scriptgeist, to basic social scenewriting in Sceneclasse and finally to more or less complete screenwriting applications in Scenewrite and Scenepad. The evolution was not done by enhancing a single deliverable from 'first cut' to 'final cut' prototype. This may have been an error of judgment on my part. As one of my supervisors (Adam Ganz) suggested, perhaps it was a mistake to deliver a number of different prototypes each with different names and web site addresses. Because by not focusing on a single prototype all the way through my research I was confusing users and forcing them to keep switching applications thereby limiting their longevity as users. This may be a factor in

relation to why the user count went down rather than up as the prototype evolved, as users did not automatically 'carryover' from one prototype to the other and became harder to convince as the functional complexity became harder to understand. Unfortunately for various reasons, including lack of developer commitment/availability forcing me to change development partner with each prototype version and availability of funding to underwrite the earlier coding effort, it was easier or necessary to start again and build from scratch with each new prototype.

FUNCTION	Scriptcloud	Scriptgeist	SceneClass	Scenewrite	Scenepad
Upload script (Text)	X	X	-	-	-
Generate word cloud	X	X	-	X	X
Generate script charts	-	X	-	X	X
Generate script entities	-	X	-	X	X
Write script scenes	-	-	X	X	X
Scene commenting	-	-	X	X	X
Scene 'liking'	-	-	X	X	X
Script storyboards	-	-	-	X	X
Script multi-level PDF	-	-	-	X	X
Script business	-	-	-	X	X
Script content hubs	-	-	-	X	X
Script scene versioning	-	-	-	X	X
Upload script (FDX)	-	-	-	-	X
Script mobile-access	-	-	-	-	X
Script sentiment	-	-	-	-	X
Script production	-	-	-	-	X

Table 1.1 – Functional evolution of prototype deliverables

In conclusion, to use Bruhn Jensen's terms (2002:237), my research design methodology can be summarized as:

- Strategy – to deliver a series of evolutionary online software prototypes to test the utility of functionality that relates to my research question.
- Tactics – to expose these prototypes to communities of interest comprising users that are most likely to engage with the prototypes.
- Techniques - to collect verbal and virtual feedback from these users and reflect on/respond to this feedback to iteratively evolve the design of the prototypes.

My first development challenge was to figure out how to 'datafy' screenplay content effectively, in order to facilitate screenplay analytics and other functionality that depends on the availability of 'screenplay as data' rather than 'screenplay as document'.

1 Screenplay as Data

We are not used to thinking of screenplays as data but rather as documents.

As Millard says in *The screenplay as prototype*, 'The screenplay is a document that exists as a carry-over from a pre-digital era' (in *Analysing the Screenplay* 2011: 146). Here, 'screenplay as data' means decomposing and digitally storing the screenplay's textual content as a collection of individual data points at a level of granularity that facilitates reassembling of these data points in various ways to present various kinds of 'wholes' that represent the sum of various 'parts': In short, the datafication of screenplay content.

Typically we think of and envision screenplays as documents. We often use them printed out on paper. There is a great deal of attention focused on the look and feel of this venerable document - use of the Courier 12 font, binding the pages using brass brads, and industry standard formatting conventions for the presentational layout of the text on the page. Screenwriting technology has largely adopted the physical paper-document format as the template digital format for a screenplay when viewed and managed on-screen therefore screenplays continue to be essentially presented, defined and discussed as document artefacts, whether paper or digital. But a document-centric artefact is not the same as a data-centric artefact.

We consider documents as artefacts designed for reading by a human 'consumer' of the document content. It is the human process of reading that is used to access the information the document communicates. So a document-centric screenplay is primarily intended for visual consumption and manual use by humans - for example to be read and annotated with a pen or highlighter and to allow additional (or revision) physical pages to be easily inserted into and 'cut' pages to be removed from, the printed script.

There are a number of ways of presenting all or parts of script content for human consumption on physical media, for example:

- As a paper document printed out for reading
- As a series of 3x5 index cards or post-it notes outlining the content of a series of scenes
- As hand-drawn boards representing a storyboard visualization of scenes

Some or all of these views of screenplay content are represented digitally in many of the current generation of screenwriting tools. The paper representation has been transitioned almost directly to a digital representation. The script has become a document format (e.g. PDF) that can be stored online and scrolled through and searched on-screen. Index cards are visually represented on-screen and display selected data about a scene. Storyboards are presented as a collection or portfolio of digital images rather than physical, hand-drawn sketches. Virtual cards and boards can be 'dragged and dropped' with a mouse to easily re-order them on screen to simulate the physical process of unpinning and repositioning them on a corkboard.

But in software development terms, a document-centric artefact concerns itself primarily with only the 'surface' presentation layer of the '3-tier' client/server architecture and design pattern – originated by John J. Donovan at the Open Environment Corporation (1994) - comprising presentation, application and data tiers.

In a 3-tier architecture, the presentation tier is what an application user sees and interacts with e.g. the user interface (UI) of a software application. The application tier contains the rules and business logic that define how the data is extracted from the data tier, how you can see what has been retrieved and what you can do with what you can see. The data tier manages the data (stored in files or a database for example) that is actually presented to the user through the lens of the user interface and as a result of the application of the business logic (for example you may only see certain data

that is based on business logic that defines your access rights according to your application user role).

A document-centric artifact is focused on the presentation or 'look and feel' layer of the content rather than on the content business rules and data and on the document as a whole, as a container of parts, rather than as a network of individual elements and their relationships. In this sense the screenplay as document places more importance on the script as whole rather than the parts of the script – the watch face rather than the components that make it function.

But a document centric artefact is not the same as a data-centric artefact.

1.1 The Data-Centric Screenplay

The document-centric screenplay is not designed or intended for machine processing. There are a number of ways of decomposing screenplay content to make the content better suited for machine (i.e. computer) processing, for example:

- As a combination of data and metadata
- As a hierarchy of relationships
- As text 'marked up' with identification 'tags'
- As text 'marked down' with specific characters to trigger specific presentation formatting rules
- As text tagged as parts of speech (i.e. dialog text identified as verbs and nouns)

A data-centric artefact is not specifically concerned with the presentation and navigation of the document as a whole. That is something that will itself be machine-processed when the data is rendered. A data centric artefact is designed both for analysis as a whole (the 'forest') and in parts (the 'trees')

and recognizes that a range of data relationships may need to be defined and maintained are drivers of, but otherwise nothing to do with the surface presentation content of the artefact but which function to facilitate its analysis and visualization in ways other than merely as an on-screen document.

You are even less likely to find screenplays discussed as the specification of a physical piece of information, which is the definition of an 'artifact' in the Unified Modeling Language (UML), the de-facto standard for building Object-Oriented software. The Object Management Group (OMG) (see www.omg.org, accessed 12 February 2014), who manage the specification of UML, state:

The Unified Modeling Language (UML) is a graphical language for visualizing, specifying, constructing, and documenting the artifacts of a software-intensive system. The UML offers a standard way to write a system's blueprints, including conceptual things such as business processes and system functions as well as concrete things such as programming language statements, database schemas, and reusable software components.

You could quite easily apply this definition to screenplays by rewriting this OMG statement as follows:

A screenplay is a textual language for visualizing, specifying, constructing, and documenting the audio-visual artifacts of a movie. The screenplay offers a standard way to write a movie's blueprints, including conceptual things such as scenes, shots and transitions as well as concrete things such as people, props and visual effects.

A screenplay uses a relatively sparse 'language' to define a means of realizing an end product – in this case a movie rather than a software application. But a screenplay is more than a document defined by a set of formatting rules that determine the way it is presented, it already comprises two core kinds of potential data content: data and metadata.

1.2 Data and Metadata

The data of a screenplay comprises only the blocks of scene description or

action and the blocks of character dialog that make up the main content of scene. Everything else – scene headings, captions, character cues, parentheticals and transitions etc. – is metadata, because it provides context for the data in terms of its on-(cinema) screen presentation. This combination of data vs. metadata is an essential characteristic of a screenplay, although it is generally unrecognized in this way. And a screenplay has considerably more metadata than say, a novel, where the only metadata is likely to be chapter numbers/headings. Novel metadata is implicit within the text rather than explicit as in a screenplay where, for example, the text is broken up by explicitly named scenes, character cues and transition instructions for example, that are not found in a novel form.

Metadata is what helps a program to understand and utilize data by providing essential context. A simple example of the power of metadata is how it is used by Disney's *Story* (see <http://story.us>, accessed 12 February 2014), an iPhone app used to create simple storylines from the masses of photos captured on an iPhone camera roll. The app uses the photo metadata (captured and stored with the photo data) to link photos together into proto-narratives or storylines. Metadata such as the date/time and location of the photo is leveraged both to group and order photos into 'starter' storylines that the user can improve and enhance using the app.

If you view a typical content page in a screenplay document you are actually viewing two separate kinds of data content: The screenplay data and the screenplay metadata that describes or contextualizes the data to give it coherence and meaning. For example, consider this snippet of dialog from *Casablanca* (1942, Michael Curtiz):

ILSA
Play it, Sam.

'Play it, Sam.' is data. 'ILSA' is metadata because it tells us something about the data (i.e. in this case, which character speaks it). A scene heading is also an example of explicit metadata in a screenplay as it provides some essential

context for all the scene content that follows.

Imagine what a given screenplay would look like with all the action and dialog blocks removed. First the script content would be more compact as the bulk of a screenplay is typically contained in the data, not the metadata. What would be left is a series of scene headings, character names and parentheticals and transition instructions. What might be called the “framework” of the script. In essence you would have the skeleton but not the organs or ‘guts’ of the narrative of a movie. Alternatively you could have just the screenplay data – comprising a continuous series of action and dialog blocks with no scene headings, character names and parentheticals and transition instructions.

One can try to film a screenplay that comprises only metadata or only data. In the first case you would be aware of specific locations and time settings, populated by specific characters but they would do and say nothing since there is no data to provide their actions and dialog. In the second case you would be able to discern some kind of narrative, it’s just that the movie could take place anywhere, at anytime with any characters since none of this is specified as the screenplay metadata is missing. Clearly it’s the combination of data and metadata that delivers a complete movie realization and potentially satisfying experience for an audience rather than simply one or the other.

But a writer writes a character name, not metadata and writes dialog, not data. As humans who understand the purpose of this content we do not need to think of it this way but machines need more help. A machine needs some explicit structure to have some indication of what the metadata or data is defined as, within the context of a screenplay, and how it should be formatted when rendered on screen/the printed page.

1.3 Unstructured vs. Structured Data

If you create a screenplay as a regular document in a standard word

processor like Microsoft Word, you are creating what is generally regarded as an 'unstructured' data artefact. Apart from a series of formatting instructions e.g. capitalize and center this text (i.e. to indicate a character cue), your word processing document has no clue what the data within the document actually means or the context it is being used in. For example if you are writing a James Bond movie screenplay in Word, your word processing software has no way of knowing that the word BOND in capital letters is in fact the name of a character in your screenplay.

In your 'unstructured' screenplay document in Word, you can search for BOND and you can use bookmarking to notate each occurrence of the word BOND in your document so that you can navigate your script content by character cue. But if, in effect, you said to your word processor 'find me every bit of dialog that BOND speaks' it would have no clue how to go about this since it neither knows what constitutes dialog in the text nor that BOND is a speaker of dialog. However, if your screenplay is stored in a structured data format you would expect a reasonably accurate response to this kind of query (if submitted in the correct query syntax of course).

To help understand the difference between unstructured and structured data in screenwriting/screenplay terms, let's consider again the dialog snippet from *Casablanca* (1942).

ILSA

Play it, Sam.

As a human reader of the snippet with my own mental contextual framework of a screenplay to reference, I understand ILSA to be a character and expect a dialog block, perhaps preceded by a parenthetical instruction, to follow. A software program used to format and analyze a screenplay must be explicitly provided with its own kind of conceptual framework in order to understand what this snippet is about. In other words, the metadata and data contained in this snippet must be further 'structured' in some way to facilitate this understanding.

Let's save the dialog snippet above in three common 'unstructured' file formats - .txt and .html and .rtf - to see how it is represented.

Format	Content
Text (.txt)	ILSA Play it, Sam.
HTML (.html)	<center>ILSA</br> Play it, Sam.</center>
Rich Text Format (.rtf)	{\rtf1\ansi\ansicpg1252\cocoartf1138\cocoasubrtf510 {\fonttbl{\f0\fswiss\fcharset0 Helvetica;} {\colortbl;\red255\green255\blue255;} \paperw11900\paperh16840\margl1440\margr1440\vieww10800\viewh8400\viewkind0 \pard\tx566\tx1133\tx1700\tx2267\tx2834\tx3401\tx3968\tx4535\tx5102\tx5669\tx6236\tx6803\pard\natural\qc \f0\b\fs24 \cf0 ILSA \b0 \br/>Play it, Sam.}

Table 1.2 – Text file representations

Other than the line return to separate the two lines, the 'plain' text file has no instructions to identify what this data is or how it should be formatted, which is why it is not centered in any way. The HTML file is 'marked' up with standard HTML formatting information for rendering this text on screen in a web browser (e.g. means show the text in boldface) using start and end 'tags' (e.g. <center> and </center>). The RTF file also includes additional 'rich' formatting information for rendering this text on screen. But none of these formats have any kind of indication of what this metadata and data

actually is. This formatting ‘style’ tagging is only used to apply formatting rules for rendering such as the font, spacing or alignment of the content encompassed by the style tags. However, here we are not concerned with presentational style tagging but with the “semantic tagging” of the screenplay content to help understand what the data actually means in its current context.

Consider if our dialog snippet was tagged in this way:

```
<character>ILSA</character>  
<dialog>Play it, Sam</dialog>
```

In this example the notation <character> or <dialog> is a “start” tag to indicate that what follows is a piece of dialog or character name and the notation </character> or </dialog> is an “end” tag that indicates the end of the extent of the dialog item. It’s easy to see how a machine that understands how to parse this kind of ‘tagged’ data can find ‘character’ data or ‘dialog’ data simply by looking for text surrounded by the ‘character’ and ‘dialog’ tags.

Traditionally, screenwriting applications recognize the need to store screenplay metadata and data in ways that facilitate correct formatting of the data when rendered. But often, to further ‘structure’ the text to better identify metadata or data for analysis purposes, this relied on manual tagging by the writer.

In order to prepare a script for production, some screenwriting software packages allow words within the text of a script to be ‘marked-up’ or ‘tagged’ by the writer or some other editor of the script with production-related metadata tags. For example, *Final Draft* provides a means for the script author to add metadata to a script by manually ‘tagging’ text in the script to identify it – say to identify a specific noun in the script as a prop. For example, references to sound, props or special effects included in the scene action description can be tagged as such:

Suddenly, the bomb <sound>EXPLODED</sound>, triggering mayhem in the crowd.

Bond whipped his <prop>WALTHER PPK</prop> from its holster.

The cop <sfx>MORPHS</sfx> into a molten metal humanoid.

The traditional formatting convention of capitalizing these production-related items in a printed-paper script is a simple way to help production managers to visually identify what was relevant to them in the screenplay to help them to plan and setup scenes/shots. Whereas manually tagging the script text in this way to 'bake-in' metadata identifiers enables the screenplay software to automatically extract specific production reports from the electronic copy of the script and to report these references in a meaningful way. The software simply searches for text within a specified tag set to find say, all the sounds, props or special effects.

This kind of tagging can also have a commercial benefit. When offered 'action' text tagged like this:

```
<action><text>Bond checked his watch: 0500 hours.</text></action>
```

A software program could specifically read only the action text in the script to find product placement opportunities for brand sponsorship to pitch to a company like Rolex by identifying the noun 'watch'. And by identifying 'checked' as a verb, the program could also start to build a profile of what the character Bond does in order automatically build up a character profile from the evidence of the data rather than by the writer describing his character manually in the form of a hand-crafted 'bio'. Does Bond do lots of checking? Maybe Bond shows signs of being a meticulous character? This is an example of how quantitative analysis of the script content may assist with qualitative assessment of a role. Now, evidence from the data helps both to

define and validate his actual functional role in the screenplay as a spy.

Tagging screenplay data in this fashion is only one way of 'datafying' the textual content, by means of embedding identification tags in the text.

Tabulating the data is another way to achieve a similar purpose.

Conceptually, a table (in practice represented by say a database table or a spreadsheet file for example) identifies the purpose of the metadata or data by defining that data in terms of a row column intersection. So (in simplistic terms), our data snippet might be represented in table format as:

Table	Row ID	Type	Character	Text
Snippet	1	Dialog	ILSA	Play it, Sam.

Table 1.3 - Row/Column Intersection

Using a database management system, the row/column intersections in the 'snippet' table can be used to query, for example:

- All ILSA's text that is of type dialog
- Only text that is type dialog
- How many times ILSA speaks any dialog
- All dialog that mentions 'Sam'
- All dialog spoken by ILSA that contains an action verb e.g. 'Play'
- All dialog spoken by ILSA that refers to another character e.g. 'Sam'

Now the metadata and data has been formally structured for identification purposes so that any program that understands that formal structure can query both in various combinations to deliver information to the user.

Once the screenplay is represented as a data-centric rather than document-centric artefact and the data structured so that programs can both identify its content purpose as well as how to format it when rendered, both data

and metadata assume an additional role above and beyond merely describing who does and says what, where and when: they help to prepare the script content for programmatic analysis: Metadata can be analyzed on its own, data can be analyzed on its own and the combination of data plus metadata can also be analyzed.

A key aim of screenwriting 2.0 applications that assume screenplays as data is to facilitate automatically recognizing semantic concepts already embedded in the screenplay content so the author does not have the burden of this additional manual process. So if the application can identify nouns used in dialog or action text because it can identify such text via metadata, the application can then extract a list of all nouns avoiding the need for the writer to tag them. This is useful because this part of speech is most likely to represent the props used in the movie's production. Counting the frequency of each noun then indicates which props will get the most use and may need to be more carefully selected.

According to some sources (see *Demystifying Big Data* p.10) only 15% of data is structured i.e. stored in databases and spreadsheets and 85% of data is unstructured in the form of emails, videos, blogs etc. (even though much of this data is actually also stored in databases). But here what we mean by 'structured' data is data that has both metadata to provide context and data that is subject a mutually agreed standard for how that metadata is defined and applied to the data What is not meant here is the typical meaning of the screenwriting term 'structure' in connection with screenplays - where what comes to mind is the Aristotelian/Field 3-act structure or Vogler's (1992) *Hero's Journey* or Murdock's (1990) *Heroine's Journey*.

There are at least three ways of storing screenplay data in a structured format to make them machine-comprehensible and more suitable for analytics: As data stored in a relational database management system (RDBMS) or in a 'NoSQL' database or in a file (or database) that uses Extensible Markup Language (XML) to 'tag' the data with metadata. I refer to these as 'tabulated', 'documented' and 'tagged' screenplays below.

1.4 Tabulated Screenplays

Instead of storing screenplay data and metadata in an 'unstructured' document file (e.g. .rtf or .pdf) or plain text file (e.g. .txt), it could be stored in a relational database management system (RDBMS) that uses a separate, formal schema to define how the data is stored and related in the database and a formal language for (among other things) querying that data for analysis purposes - Structured Query Language (SQL). This approach has been used before for a similar purpose, for example by Franzosi (2010: 75-82), to store and analyze what he calls the 'story grammar' of any kind of narrative text in a Microsoft Access application called PC-ACE.

Here, the schema (or entity-relationship model) defines (among other things) the database tables used to represent the main data entities of the script and the table columns used to store both data and metadata within those tables.

Simplistically, a screenplay could be represented by just 3 related database tables (see figure 1.1):

1. A script table
2. A scene table
3. A 'snippet' table (storing action and dialog data)

The script table has one row per script. The scene table has one row per scene within a script. The snippet table has one row per data element (e.g. action or dialog block) within a scene.

These tables are related by means of 'foreign keys'. In relational terms, if the primary key of the script table is `script_id` then this is used as a foreign key in the scene table to link one or more scene rows to a script row. If the primary key of the scene table is `scene_id` then this is used as a foreign key in the snippet table to link one or more snippet rows to scene rows.

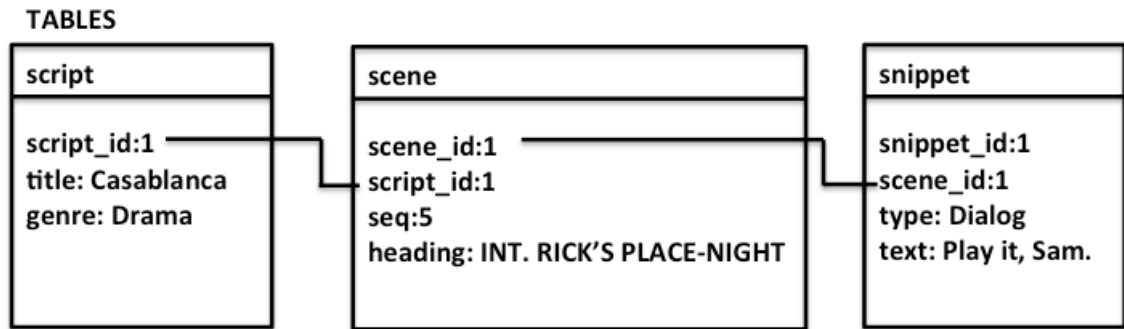


Figure 1.2 - Script as tables (simplified)

So a single screenplay with 60 scenes and many action and dialog blocks within those scenes is likely to be stored as:

- 1 row in the script table
- 60 rows in the scene table
- Hundreds of rows in the snippet table

Most of the 'data' stored in these tables is in fact metadata. The snippet table stores both, with a row comprising a series of columns in which data is represented by, for example, the dialog text of the snippet e.g. 'Play it, Sam' whereas metadata is represented by, for example, the type of snippet represented e.g. 'Dialog' or an identifier of the character who spoke it.

By tabulating the screenplay data and metadata in this way, the screenplay is stored in a structured format that includes formally defined relationships between data tables. To find out information about a script e.g. the title or author, you query the script table. To find out information about a scene, e.g. the location or time of day you query the scene table for a specific script. To find out information about the dialog or action in a script scene, you query the snippet table. To get a copy of the complete script document, that requires data from all three tables, you would need to assemble (join) data sourced from each table based the relationships between them.

All my software prototypes (described below) are based on tabulating the data in the Open Source RDBMS MySQL. The main advantage of tabulating

screenplays in this way is that it makes deriving analytics from the data much easier as there are many tools available to analyze and visualize data stored in SQL databases. A potential disadvantage is that all scenes and the script as a whole have to be assembled 'on-demand' from the scenes and the snippets linked to each scene. Another is that as the script is no longer encapsulated within a single document, but decomposed into smaller content units, it requires more formal management and may not present the traditional free-format document 'feel' to users.

Franzosi (2010: chapter 3) also makes a number of useful points about using an RDBMS to store data and using SQL not just to query the data but also to update the data to better reflect the analytical needs (2010:93-95). So if, for example, you wanted to identify all violent scenes in a screenplay (without manually doing this yourself) you could add a column to your scene table – say `scene_type` – and then use a SQL update statement to find examples of terms in the textual data of the scene (i.e. dialog and action snippets) that can be 'mapped' to the `scene_type` attribute 'violent'. For example the terms 'killed', 'stabbed', 'shot' can be identified in the text and the scene flagged as `scene_type` 'violent'. A similar data update process could also be used to identify sexual scenes from the screenplay text. The combination of these 2 techniques could identify scripts likely to have 'adult' content without the need to read the script.

1.5 Documented Screenplays

Recently, a number of 'NoSQL' databases have emerged that store and manage data differently from a SQL RDBMS (notably they do not use SQL as the language for managing and querying the database) and are better suited to managing specific kinds of textual content. Vaish (2013:11) identifies five types of NoSQL database, segregated by data model:

1. Document
2. Key-Value
3. XML

4. Column

5. Graph

My aim here is not to compare and contrast these 5 types but simply to say that while any of these 5 types could be used to store screenplay data, some are more likely candidates than others due to the relatively stable structure of a screenplay and the nature of its content. For example, NoSQL graph databases (like Neo4J or FlockDB for example) are best suited for storing only that data from a screenplay that is specifically intended to produce social network graphs (e.g. to visualize the character relationships in a screenplay).

Here, as a contrast to the SQL RDBMS or 'tabulated' datafication approach outlined above, I will outline the use of a NoSQL Document database (like Couchbase – see <http://www.couchbase.com/docs/couchbase-devguide-2.0/modeling-documents.html> - accessed 10 June 2013) to store screenplay data.

Document-databases have become popular as a data store for web applications such as social networks where the data being managed is predominantly high volumes of textual (vs. say numeric) content such as comments or chat or tweet-like data. Instead of requiring a pre-defined schema to structure the data into tables comprising a set of columns with data represented and accessed as content rows, document databases store data in document entities with the data stored in value pairs. So separate documents could represent a script, a scene and a snippet (see figure 1.3).

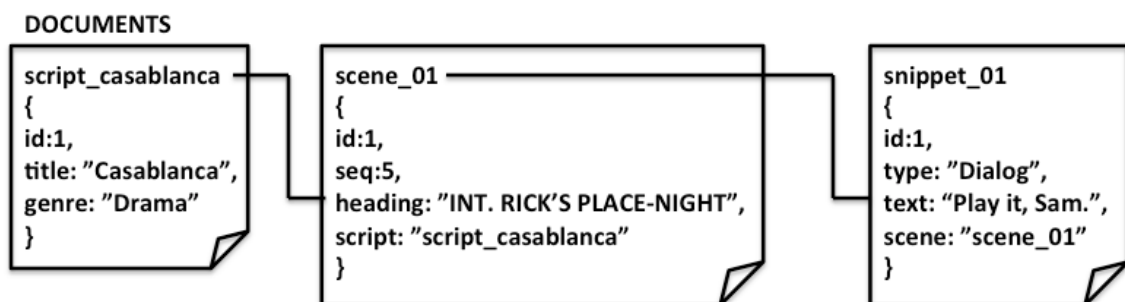


Figure 1.3 - Script as documents (simplified)

Here the document is like a table, the value pairs like columns and the linking relationship between documents is based on a document reference identifier rather than primary and foreign key references in the relational model.

Document databases are claimed to have some advantages over the relational model in that they allow easier distribution of these virtual documents across physical servers, require less schema administration (because the document itself is its own schema) and offer more efficient data updating if data changes regularly (as it is likely to do while drafting versions of a screenplay). Documents can also include linked data 'nested' within a single document - such as a scene document that also contains all the comments posted about that scene - so that from a content perspective the document is fully integrated and can 'stand alone' for analysis purposes. In the relational model this would require two related tables: one for scenes and one for comments linked to those scenes. A disadvantage of these NoSQL databases over SQL databases is that they are newer - my lack of knowledge of these databases is the main reason why I did not use this kind of data store for my practical deliverable Scenepad - and they are not as accessible from as wide a range of analytic and visualization tools as SQL databases. However it seems highly likely that NoSQL databases will be used by future screenwriting 2.0 applications.

1.6 Tagged Screenplays

Instead of storing screenplay data and metadata in a database, the data could be stored in an Extensible Markup Language (XML) file. As mentioned above, this is the basis for the file format now used by Final Draft, so this type of data format is already in widespread use. Note that Final Draft stores the XML-tagged screenplay data in a discrete, named file and not in a database designed to store XML-tagged data where both the data and the tags are stored in the database itself.

However, XML databases can also be used to store XML-tagged screenplay data in a database management system rather than in a file. In this case, each individual 'instance' file is fed into the database and the tagged content 'shredded' into the more granular data storage structure of the database. An XML database would be the ideal repository for managing a corpus of multiple Final Draft .fdx files because the power of the database query tools can be brought to bear on the corpus as a whole rather than having to work with each screenplay file-by-file as is the case when each screenplay is 'stored' individually in its own .fdx file. Creating such a corpus of .fdx screenplays in an XML database is a significant research and commercial opportunity.

When screenplay data is stored in XML it functions as an XML 'instance' document that is part of a related set of documents that typically includes both schema (.xsd) and stylesheet (.xsl) files (and may include others) used for defining the content and presentation of the instance document as a specific kind of artefact within a specific information domain (in this case a screenplay artefact within the screenwriting domain). The tags used to markup the screenplay metadata and data content, are defined in the schema and their rendering/presentation format described in the stylesheet. So if a *Casablanca* screenplay were stored in this way it would probably consist of at least 3 files:

- casablanca.xml – the data/metadata instance document
- casablanca.xsd – the XML 'reference' schema document
- casablanca.xsl – the stylesheet document

The screenwriting application would include a 'parser' function that is able to read and format the content of the Casablanca.xml file with reference to the xsd and xsl files. Note that a Final Draft .fdx file combines all the screenplay data and metadata and XML tags, including the relevant presentational attributes, into a single file and it is the Final Draft application itself that interprets and presents this XML file to the user.

XML is not only useful for providing a means to structure the data, but also has a potential standardization purpose that could benefit the screenwriting fraternity in other ways. For example, if a 'standard' XML format can be agreed for describing screenplays, in the form of an agreed schema and presentational stylesheet (as has happened in other business domains – see section 1.6 below), then it will become much easier both to export/import a homogenous screenplay format between heterogeneous screenwriting applications and to apply similar analyses and visualizations from a single screenplay analytics application across content sourced from multiple screenwriting applications.

A disadvantage of screenplay tagging is that the addition of markup data to the script will significantly increase the size of the screenplay file reflecting the addition of start and end markup tags to the text. But today's cheap storage and processing power makes this irrelevant and this does not matter because the marked up screenplay file is not a text for reading by humans (although some may find this an interesting pursuit), but for processing by a software program.

XML is a specific way to tag data and metadata but 'tagging' can be done in other ways. For example Franzosi (2010:62-66) describes a user manually 'assigning' 'codes' to text using the Computer-Assisted Qualitative Data Analysis Software (CAQDAS) ATLAS to add the metadata attributes. He describes assigning the code 'Participant:Actor:Fascists' to the term 'fascists' in a text in order to identify the subject of a subject-verb-object triplet to facilitate further qualitative textual analysis. Once all text is assigned codes manually in this way (see *ibid*: figure 3.20) it becomes relatively simple to query the database to retrieve examples or a count of all 'actors' in the text (*ibid*:66).

XML tagging can also be used to model a hierarchy, by encapsulating one set of tags within another. An example of a screenplay schema model that reflects a construct of XML element tags and attributes is shown in figure 1.4 below. The <script> tags encapsulate the <sequence> and <scene> tags and

the <scene> tags encapsulate <action> and <dialog> tags. The term “Text” refers to the actual text of a specific dialog or action block instance in the script.

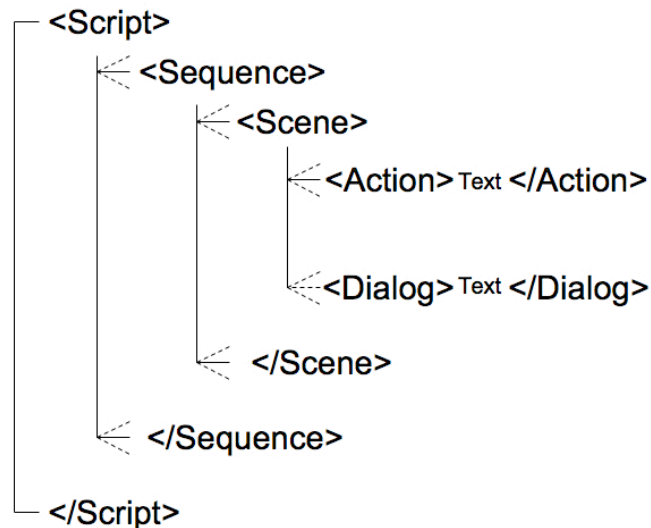


Figure 1.4 - An Example of a Screenplay Hierarchy Model

An XML tag (or element) may also be associated with and qualified by metadata of its own, so-called “attributes” (not included in figure 1.4) used to uniquely identify a specific instance of tagged data in some way, so for example:

- The attributes of the <script> tag may be: Title, author and date.
- The attributes of the <sequence> tag may be: Title and order (i.e. each sequence has its own title and running order in the script).
- The attributes of the <scene> tag may be: INT/EXT, location, DAY/NIGHT, transition instruction and order (i.e. each scene has its own heading/slugline and running order in the script (or sequence) and potentially a scene-end transition instruction).
- The attributes of the <dialog> tag may be: Character, parenthetical, transition instruction and order (i.e. each dialog block is spoken by a character in a particular way and in a

specific order within the scene with potentially its own transition instruction).

- The attributes of the <action> tag may be: Transition instruction and order (i.e. each action block has a specific running order in the scene and potentially its own transition instruction).

Tagging may be an appropriate structured data format for screenplays since it has already been used to bring the benefits of structured data and standardization to other business domains focused on textual content, for example corporate regulatory reporting.

1.7 XBRL and Corporate Reporting

To understand the potential of screenplay as data, it helps to compare the screenplay to another kind of document content. I have chosen the mandatory corporate reporting submitted by businesses to regulators in business jurisdictions around the world. This kind of reporting often takes the form of quarterly or annual 'returns' such as the 10-Q and 10-K reports submitted by registered businesses to the Securities and Exchange Commission (S.E.C.) in the USA who mandated the use of Extensible Business Reporting Language (XBRL) submissions of the 10-Q and 10-K reports by corporate filers starting in 2009. XBRL is a specific implementation of XML used to 'tag' these reports so that most of the data that was previously reported in an 'unstructured' way (as an HTML document) in the past now has to be provided as both an unstructured HTML document file and a structured XBRL data file.

Until very recently most businesses filed their corporate returns to regulators around the world either as paper/PDF/HTML documents or by filling in an online form – a digital representation of the paper document. Unfortunately this kind of delivery of corporate data is not conducive to the kind of analysis that government and other regulators need to make of company data (e.g. to quickly identify trends or fraudulent activity or simply

arithmetic errors) and the like-for-like analysis that the investor community needs to make decisions about whether or not to invest in a company. The reason for this is that the data may not be consistent, is subject to transposition errors as data is 'taken off' documents and re-keyed into the databases or spreadsheet models or that one filer's revenue or expense calculations are not made on the same basis as those of others because the data submission was not structured and standardized in a mutually agreed way (i.e. mutually agreed by both the producers and consumers of the data).

Regulators and financial services investment analysts that consume the data produced by corporates in their returns are using the data for serious purposes – for example to detect financial fraud or to make influential equity 'buy', 'hold' or 'sell' recommendations to the brokers and individual investors that subscribe to their services. Some of the benefits of XBRL submission are claimed to be faster processing of the submissions themselves and of analysis of the submission content and greater reliability of the data submitted as to provide the kind of comparability that company regulators and investors need to analyze company performance efficiently and effectively requires a more rigorous data delivery.

This is why many regulators around the world, including HMRC in the UK and the S.E.C. in the USA, have mandated that certain quarterly and annual corporate reporting is provided in a specific format based on Extensible Business Reporting Language (XBRL). XBRL is used to 'markup' reporting data according to a specific data standard in the form of a mandated XBRL schema or taxonomy that all filers must use. The data can still be presented in the traditional way but the surface presentation now has the potential for analytical depth because the data structure is more rigorous, based on a single standard being used to define data concepts, rules and relationships (e.g. formulas) as reflected in the data markup.

The data markup, in this case XBRL tags, is subject to a regulator-mandated schema. This ensures that the analysis programs used by both regulators and other information consumers, can expect to compare like-for-like tagged

data to look for exceptions, to identify patterns and trends or to create summary aggregations based on mutually agreed formulas also defined within the shared and mutually agreed XBRL files referenced by these programs.

Appendix A outlines a very simple example of using XBRL as provided by Charles Hoffman (aka the 'father' of XBRL) for learning purposes – see <http://xbrl.squarespace.com/journal/2008/12/18/hello-world-xbrl-example.html> - accessed 6 June 2013). Note that all data content is enclosed within <start> and </end > tags.

In this case, the instance document *HelloWorld.xml* relies on the *HelloWorld.xsd* schema document for its data processing logic so the schema acts as the key reference for any program that is reading the instance document. In practice this means that if many businesses create instance files to report data that refer to the same schema then there is a data consistency and standardization in play that makes the data derived from processing the instance document more likely to be reliable and comparisons more accurate. In the case of XBRL reporting to the S.E.C., the S.E.C., in its role as information regulator, 'owns', maintains and enforces the schema that all corporate information providers utilize.

Corporate reports submitted to HMRC or the S.E.C. for example are much more complex instance files and subject to much more complex schemas than this HelloWorld example, but conceptually they function in the same way.

By submitting the corporate reporting data in this XBRL format, the traditional return document has been datafied so that each filing can be delivered as an electronic upload that is received, processed and 'first pass' analyzed entirely by machine. In the case of S.E.C. XBRL submissions one immediate benefit is that the corporate submission data is also available virtually immediately for public consumption via an RSS feed that is also generated automatically from the submission file, programmatically.

However, a screenplay is not a corporate report and there is no mandate to create screenplays using a specific data format, nor is there an agreed XBRL schema for screenplays (although there might be one day). But screenplays could be treated in the same way given that agents, movie studios and production companies that receive 'spec' scripts for evaluation essentially play a similar content-receiver role to a corporate regulator and it could be advantageous for all such receivers to receive their screenplay data in the same format. But this is likely to be some years off yet.

As yet, unlike the corporate reporting domain, there is no standard file format or XML schema for sharing screenplay content despite the efforts of individuals such as Sean Moubry (see <http://screenplayXML.org> – accessed 10 June 2013). However, with the transition of Final Draft from the proprietary .fdr format to the XML-based .fdx format (in version 8 released April 2009) and Adobe *Story*'s story XML format, there is now real industry momentum towards XML as the file format of choice. For example, other screenwriting applications like ScriptsPro (for the iPad/iPhone) have already adopted the Final Draft .fdx format as their default storage format.

Final Draft .FDX files use 'paragraph' tags to identify screenplay data/metadata and also to help drive formatting of the data when presented on-screen or on-paper. The example below from the fdx file of a version of the script of *Alien* (1979) begins with a <Content> tag, to define the start of the script, and uses the <Paragraph Type =""> to identify the different screenplay elements (e.g. Transition, Scene Heading and Action below) and to drive the content formatting.

(This example also shows an error in the *Final Draft* file import logic as the caption 'SOMETIME IN THE FUTURE' has been incorrectly defined as a transition element because it ended in a colon, which the import business rules assume is a signifier of a transition.)

<Content>

<Paragraph Type="Transition">

```
<Text>FADE IN:</Text>
</Paragraph>
<Paragraph Type="Transition">
<Text>SOMETIME IN THE FUTURE:</Text>
</Paragraph>
<Paragraph Type="Scene Heading">
  <SceneProperties Length="1/8" Page="1" Title="" />
<Text>INT. ENGINE ROOM</Text>
</Paragraph>
<Paragraph Type="Action">
<Text>Empty, cavernous.</Text>
</Paragraph>
```

Other options gaining some attention are Kent Tessman's Open Screenplay Format (<http://www.kenttessman.com/2012/02/open-screenplay-format>, accessed 28 August, 2012) used in some recently released screenwriting applications (like Tessman's own *Fade In*) and John August's Fountain format (<http://www.fountain.io>, accessed 29 March, 2013). These file formats are designed for use by any screenwriting application and therefore less tied to the proprietary internals and needs of specific applications like *Final Draft* or *Story*.

1.8 Analytic Metadata for Screenwriters

Production-related markup of the screenplay text (e.g. for props or sounds) is directed at a specific set of script stakeholders, namely the various production managers. It is not intended to help the screenwriter create a higher quality, more saleable script. Markup to help the writer improve their script is lacking in almost all screenwriting software.

For example, character names are an important piece of metadata in a script. They are linked to sections of dialog, and may also be mentioned both in the dialog of others and in scene action descriptions. But to get any analytical value from character-related data/metadata in the script means that it must

be tagged everywhere it occurs in a script. In the screenplay of *Casablanca*, the name “RICK” must be tagged as an element-type “character” everywhere it is found, in order to define instances of RICK in the script content as representing the character Rick (and not say a hayrick), for example:

```
<character>RICK</character>
<dialog>I stick my neck out for nobody</dialog>

<action><character>RICK</character> moves out of the
        shadows and into view.</action>
```

Alternatively, if these dialog and action snippets are stored separately as rows in a database table then the database row itself can be linked relationally to the character ‘Rick’ (itself a row in another table) so we know it was ‘character’ Rick who spoke or ‘character’ Rick who was in the action. Either way, once Rick and the other characters in the script have been identified, it is realistic to expect a program capable of reading both the screenplay data and metadata to determine the following:

- what dialog Rick speaks
- who Rick speaks to
- when Rick is referred to by other characters
- which scenes Rick plays a role in
- which action Rick is involved in or referred to in

As a minimum, knowing what dialog Rick speaks enables dialog analysis to produce some kind of linguistic fingerprint for Rick; knowing which characters Rick speaks to and is spoken of by, clarifies the relationship network that Rick participates in and knowing which scenes Rick plays a role in or which action he is involved in helps to establish both his narrative “throughline” and character “arc” in the screenplay.

Now imagine that Rick is identified not just as a character but that his character is also separately described by further, enhancing metadata

“attributes” such as role (e.g. protagonist or antagonist), gender, race, sexual preference etc. This opens up even more analytic potential by expanding “existing” tags, intrinsic to the screenplay, with the addition of “enhanced” tags added later by the writer/analyst. Clearly generating metadata tags from and adding metadata tags to existing script content is an important value-add process in terms of maximizing the analytic and visualization potential of the screenplay content. This is the kind of metadata that my *Scenepad* prototype allows.

1.9 Screenplay Elements

The XML/XBRL concept of ‘elements’ (that define the tags used to identify data/metadata in content) can easily be applied to screenplays as a means of understanding how screenplay content could be datafied. *Final Draft* uses the term ‘elements’ to define the list of formatting options for the screenplay’s text including: General, Scene Heading, Action, Character, Parenthetical, Dialogue, Transition, Shot and Cast List. You type in the text and pre- or post-apply the element ‘tag’ to format the text appropriately.

For analysis purposes, it’s also useful for the content of a screenplay to be viewed as consisting of an assembly of individual, but related, textual elements. In *How to Be Your Own Script Doctor* Kenning (2006:6) states that:

Analyzing a screenplay is an intricate process that is the same for any document – it entails breaking apart the individual elements and relationally subjecting those elements to each other and to the whole.

Although she refers mainly to higher-level (or lower granularity) elements than I discuss below, it is important to recognize that a screenplay can be approached as a construction of a series of inter-related elements.

In a screenplay, each element has both a content purpose and a specific formatting convention to enable the element to be viewed correctly (i.e. to industry standards) when rendered on-screen or printed on paper. In the brief discussion of screenplay elements that follows, standard script-formatting conventions are neither discussed nor complied with in the

examples, only the purpose of the element is outlined. Some of the examples below are based on a version of the script for *Casablanca* (1942). A visual overview of various typical screenplay elements is shown in figure 1.5 (see <http://steveboese.squarespace.com/storage/sample-screenplay-page.gif> - accessed 7 June 2013).

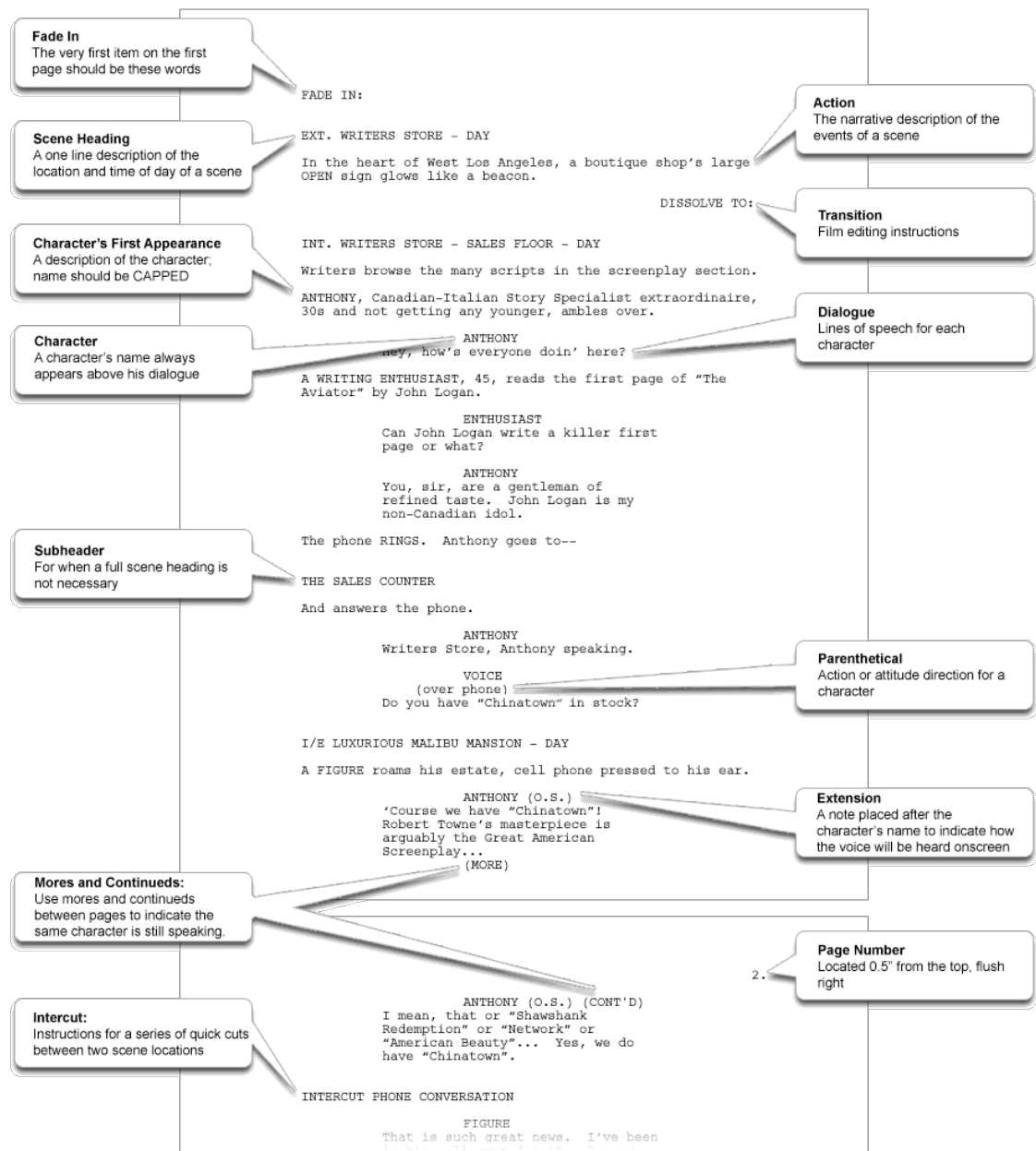


Figure 1.5 – Typical Screenplay Elements

Fade In/Fade Out

These terms are often used to start and end a screenplay, and if included they act as markers as to the extent of the script content. A screenplay file may also include a title page and possibly other front matter content relating to what might be seen while initial the movie credits are showing, but the first FADE IN usually indicates where the actual screenplay content starts. Similarly, FADE OUT or END would indicate where the screenplay ends. If these extent terms are missing, then it can be assumed that the screenplay starts when the first identifiable scene heading occurs and ends when the last identifiable scene finishes.

The Scene

The primary building block of a screenplay is the scene, which usually combines both dialog and action taking place at a specific internal or external location during a specific time period at a specific time of day. Scenes may be intentionally grouped into contiguous dramatic units (sequences) or into not necessarily-contiguous dramatic units (storylines) and may be also be deliberately written to facilitate the decomposition of the scene content into individual “shot-lists” for use during production.

A scene typically comprises:

- The scene heading
- One or more blocks of scene and/or action description
- One or more blocks of dialog and/or voiceover narration
- One or more audio/visual transition instructions occurring either at the end of the scene or within the scene

Scenes have an identifiable beginning in the script content. Conventionally,

all scenes should begin with a scene heading (starting with INT or EXT). Some scenes end with a transition instruction (e.g. CUT TO). Dialog begins with the cue or name of the character speaking the dialog. These start signifiers provide obvious entry points for some kind of programmatic script content analysis method.

Scene Heading (or “Slugline”) and Captions

A screenplay consists of a series of scenes that may or may not be in chronological order in story terms. Each scene has a heading, for example INT. RICK’S PLACE - NIGHT, which consists of at least three sub-elements:

1. INT. or EXT. - indicating whether the scene will be shot internally (e.g. in a room) or externally (e.g. on a street).
2. LOCATION - a brief description of where the scene takes place (e.g. which room in a house or a physical place).
3. DAY or NIGHT – a time of day indicating whether the scene takes place sometime during the day or sometime during the night.

Occasionally a scene heading may be extended to contain extra information such as the nature of the scene and/or a specific time of day, the season, present/past/future setting - for example: EXT. THE AFRICAN SAVANNAH - DAY – PREHISTORIC TIMES (DAWN). It may also be followed by a caption that establishes some exact criteria for the scene location and time that adds essential contextual information for the scene e.g. The White House – The Morning of 9/11.

Action/Scene Description

A scene usually comprises both action (or scene directions) and dialog (or spoken parts) but could comprise only one or the other. An action-only scene is quite common. However, a dialog scene without any action is unlikely, since some kind of scene description is usually essential to

establish a context for the dialog to take place.

Action descriptions may or may not involve naming any of the characters speaking or being spoken to in the scene but will include the mention of props, special effects (SFX), specific sounds and other production-related items. These character names and items may be capitalized in the script as this used to be a way of highlighting them within the textual content of a printed document for production planning purposes. Today, if scripts are viewed online, there is less need to capitalize since these keywords can be color-coded based on user-defined tagging.

Character and Dialog

If there is dialog in the scene, it is spoken by a character (even if that character is invisible to the camera e.g. a narrator providing a voiceover), for example:

ILSA
Play it, Sam.

The character would usually speak "on camera" (i.e. within the frame of the shot) but may speak off-camera (O.C.), off-screen (O.S.) or as a voice-over (V.O.). Using a parenthetical, the screenwriter may also provide a specific suggestion to the actor in relation to the dialog that follows, for example:

RICK
(indifferently)
If I gave you any thought, I probably would.

The first time a significant character is mentioned in the script, their name is capitalized – traditionally an indication to the actor of when their part “starts” in the script. Again, today, this convention is hardly necessary as it is easy to find the first mention of a character in an online script through a search function.

Transitions

A transition (e.g. CUT TO: or DISSOLVE TO:) moves the focus of the audience's attention to create a specific effect within the visual flow of the movie's narrative. A transition can occur either within a scene, say to cut to and from different locations to set the context for a telephone conversation, or at the end of a scene to move to the next scene in a specific visual way.

Transitions affect both the look and feel and the pacing of a movie. However it should be noted that transitions written into the script by the screenwriter may be ignored altogether in production and in any case have much less impact on the final product than the shot-by-shot transitions decided upon by the movie's camera director during production or the editor, post-production.

Other Elements

A screenplay, created in screenwriting software that supports this capability, could also include structural act and/or sequence headings inserted into the script and sequential scene numbers. These headings are not part of the screenplay content as such, but function to group scenes together for script organization and locate scenes for script navigation. An act is a collection of sequences or scenes (if sequences are not used) and a sequence is a collection of scenes that comprise a deliberate unit designed to achieve a specific dramatic or narrative purpose.

Where dialog is broken up by action or a page-break, the terms MORE or CONT'D may be inserted to indicate a break in the flow of the dialog content across printed pages.

The screenplay may also include specific camera directions such as ANGLE ON (i.e. shot angle) or CLOSE UP (i.e. shot framing), although this is less common today than it used to be. And although more common in TV scripts, a storyline identifier is also another way of grouping scenes, In this case, scenes that belong to a specific narrative thread in the screenplay.

A screenplay is a dynamic artefact so during the production process, revisions to the script may also be added and provided on different coloured paper sheets, these revision pages signifying a new 'version' of the previous script content.

Screenplay Text

Arguably, the lowest level of meaningful granularity in a screenplay is the individual word (contained in action description or dialog), but the lowest level of element granularity is the blocks of dialog or action description in the screenplay. This is the true 'text' of the screenplay. This text is analogous to what Florian Korsakow's Korsakow system (see <http://korsakow.org>) calls the '*smallest narrative unit*' (SNU) - although he uses it to refer to short video clips that can be combined in his 'dynamic storytelling' system to create new storylines.

The text comprises individual words and the syntactic units e.g. sentences that organize the words into units of meaning, both of which are potential subjects for textual analysis. Analysis of words and word frequency may an indicator of a number of 'traits' of the script as a whole: time period, location, mood, theme etc. Sentence structure and words grouped into terms may also be indicative of time period, milieu, culture, pace etc. Either or both can be used to determine role or scene characteristics such as positive or negative sentiment, profane or sexual content, or assist with opportunities for product placement.

The text can be searched for keywords and event notifiers such as discourse markers to identify events within a text that may then indicate something else, such as some kind of 'change' in the narrative that is preceded by the event marker.

1.10 Screenplay Decomposition

Other than as elements, a screenplay can be decomposed in a number of ways, including as a hierarchy, a set of entities, parts of speech, and a series

of content strata.

Hierarchy Decomposition

As a hierarchy, a screenplay may be decomposed as follows:

Acts

Sequences

Scenes

Action(s)

Transition

Character(s)

Parenthetical

Dialog

Transition

Transition

This hierarchy provides a built-in "tree" structure for navigating the script content and to allow content viewers to easily "jump-to" specific content locations in the script to read or edit the textual content. As noted above, XML supports the "nesting" of tags within tags and so XML is very suitable for representing this kind of built-in tree. Most screenwriting software already provides rudimentary hierarchy navigation, if only in the form of a list of scenes that lets a user click the scene name in the list to view/edit the full scene content.

However, a screenplay does not just have one hierarchy as it depends on the point-of-view, which in turn sets a 'start point' for the hierarchy. In the

example above the start point is the script. But if the start point is a specific role or location then the hierarchy will be different, for example to navigate only the dialog snippets linked to a role or the action snippets linked to a location.

Entity Decomposition

Scripts may also be viewed as a collection of "entities" – an entity being an attribute with meaning. A 'character' is an entity since it has a specific meaning in terms of its role in the script narrative. A 'location' is an entity since it has a specific meaning in terms of where the narrative takes place. An entity may be an example of a specific implementation or "instance" of a certain higher-level entity "class": For example, characters and scene locations as examples of screenplay entities reflecting screenplay-specific implementations of the general entity classes of "people" and "places". Blocks of action can also be considered as screenplay specific entities representative of a general entity class of "events". A block of action may be further decomposed into other sub-entities such as place names, prop names, sounds, special effects (SFX) and so on. Transitions are also entities, each transition being a specific implementation of the "visual transition" class.

Chatman (1980:26) produced a form of entity-class view of narrative that is relevant here, and divided it into two main branches: Story (Content) and Discourse (Expression). For our purposes, the branch "Story" leads to the important sub-class branches of Events (action and happenings) and Existents (characters and settings cf. characters and locations above).

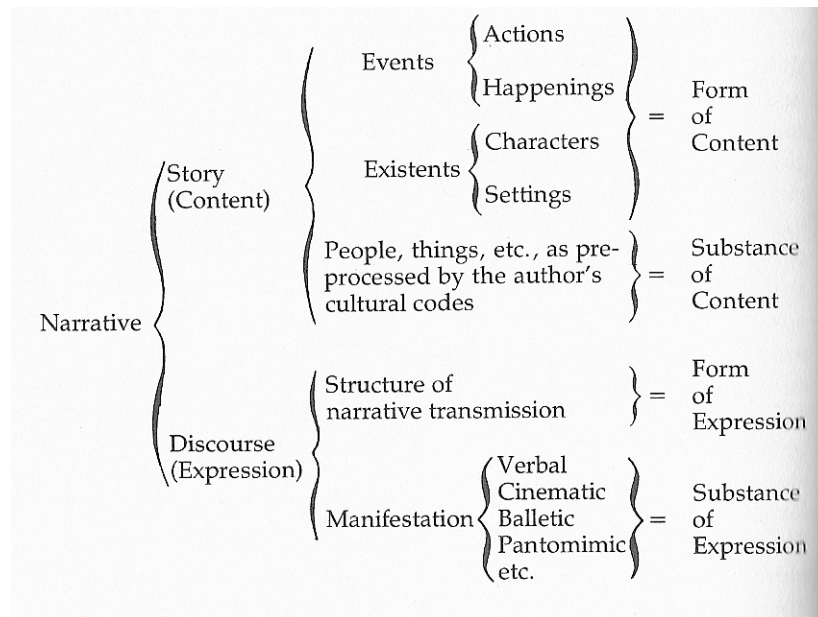


Figure 1.6 - Chatman's Narrative Entity Classes

Linguistic Decomposition

As Chatman indicates, a block of action or dialog in a screenplay also represents a form of expression of a specific kind of Discourse entity. Here the form may be further decomposed linguistically, for example into parts of speech (e.g. verbs and adverbs, nouns and adjectives) and significant keywords, such as any character names involved in the action. In the case of dialog, linguistic decomposition may be used in order to determine some kind of tonal fingerprint (e.g. emotional level or the use of sexual/profane language) for the content analyzed. The parenthetical and voiceover/off-camera indicators may indicate vocal loudness, auditory focus or a character's "distance" from the scene as we see it (e.g. a voiceover may imply that the character is "looking back" over events that have already happened). Repetitive use of certain keywords in the dialog or action throughout the script – for example the use of the word 'family' in the *Godfather* (1972) script – may be indicative of a core theme in the screenplay (see Scriptcloud prototype in 'Road to Scenepad' below).

Strata Decomposition

But this kind of entity decomposition is based on the obvious "surface" content of the screenplay, namely the words themselves and their structural organization in a screenplay format. Another kind of decomposition seeks to understand what might be termed the underlying "strata" of the screenplay that may or may not be explicitly found or visible in the textual surface.

In *Narrative Fiction*, Rimmon-Kenan (2002:10) expresses it this way, 'Whereas the surface structure of the story is syntagmatic, i.e. governed by temporal and causal principles, the deep structure is paradigmatic, based on static logical relations among the elements...'

Uncovering these logical relations to reveal the "deep structure" is the challenge here. Metaphorically, the words and sentences represent the visible landscape, whereas the strata represent the invisible geological layers below. These strata could take the form of:

- Timelines
- Journeys
- Arcs of change (i.e. changes in emotional or psychological states)
- Tones (e.g. colours, temperature)
- Sound and Silence
- Seasonal Variations
- Past, Present and Future
- Symbols and Metaphors
- Plots and Sub Plots
- Narrative conventions

- Themes
- Pacing

Unlike say screenplay structure, which represents a ‘fixed’ overlay of the content, these strata are more fluid and more likely to be impacted by even small changes to the script content. There may also be certain strata patterns that are particularly relevant to the conventions of specific movie genres such as those relating to death and violence in horror movies or sex and love in romantic comedies. Strata analysis and visualization is likely to demand relatively sophisticated algorithms to detect, decode and display the strata visually to the writer after taking the script’s proposed genre into account.

Resource Description Framework (RDF)

According to Tauberer in *What is RDF* (2006:1), ‘Resource Description Framework (RDF) is the W3C standard for encoding knowledge’ and a key part of the future semantic web, which he terms ‘a decentralized platform for distributed knowledge’. RDF is a method of decomposing knowledge into individual facts, expressed as a Subject-Predicate-Object combination termed a “triple”. Among other things, these facts can also be associated with a Uniform Resource Identifier (URI) i.e. a web link and with text value attributes (literal values). For example, a URI could be a web URL that locates the triple on a specific web page and a literal value could be a descriptor that tells us something more about the triple, e.g. to add specific context or additional content to the triple.

It is quite practical to decompose a screenplay into a large number of triples that would summarize a number of facts about the script, for example:

<scriptname>is written by<author>

<scriptname>belongs to<genre>

<scriptname>is WGA registered as<registration identifier>

<character>speaks<dialog>
<character>speaks to<character>
<character>takes part in<action>

<action>requires<special effect>
<action>involves<propname>

<dialog>is<narrated>
<dialog>is spoken in<language>

<scene>is located at<location>
<scene>takes place at<day/night>
<scene>belongs to<sequence>

Decomposing a single screenplay into RDF doesn't necessarily make that much sense, but if a corpus of screenplays is decomposed in this way and made accessible via the Internet, this corpus of screenplay facts in RDF format could be queried in ways that may result in interesting analyses or visualizations.

1.11 Screenplays as 'Big Data'

It's difficult to discuss screenplays as data without some reference to the data buzzword 'du jour', namely 'big data'. On its own, a screenplay hardly qualifies as 'big data' – the term usually used to refer to the high-volume and high-velocity datasets that governments, retailers and banks routinely deal with for example. And a single screenplay does not represent anywhere near the kind of data variety, velocity and volume that big data algorithms are designed to thrive on – for example to discover data patterns.

Big data is generally created by large numbers of people or devices 'transacting' with data collection systems on a regular basis. In this sense, the big data is created by 'pushing' data into a data collection system. For example when each bar coded product that you buy at a supermarket is logged into a centralized point-of-sale system so the data can (among other

things) be used to determine optimum inventory levels by store and across a chain based on actual and predicated spending or to target offers to consumers based on past spending habits.

Big data can also be collected by crowdsourcing data posted by users to online data streams. In this sense the big data is created by 'pulling' data into a data analysis system. For example when Twitter Tweets or Facebook Likes are used to perform sentiment analysis to look for trends that may indicate increasing or reducing interest in an event, person or product.

Compared to these 'push' and 'pull' examples, an individual screenplay represents merely a tiny sample of a bigger data picture, like a pixel in a digital photo, that could be represented by a corpus of screenplays representative of a genre, a director's body of work over a long career or all scripts from a movie-making decade for example. That's why the initial interest in applying 'big data' techniques to the world of movie making has not focused on individual screenplay content analysis to improve script quality but in activities like trying to predict the best films to acquire for optimum online rentals or which movies could become box-office blockbusters or potential Oscar winners.

The *New York Times* reported in *Giving Viewers What They Want* (David Carr – 24 February 2013) that online movie rental giant Netflix analyzes big data sourced from their customer activity to refine their ability to decide which shows to license for rental. This big data includes some 30 million movie 'plays' per day, over 4 million subscriber ratings, 3 million searches plus other data relating to when subscribers watch shows and how they pause, rewind and forward shows.

The *New York Times* also reported on the commercial script analysis activities of the *Worldwide Motion Pictures Group* in *Solving Equation of a Hit Film Script, With Data* (Brook Barnes – 5 May 2013). Screenplay analysis of specific scripts, combined with other data from focus groups and online surveys is supposed to help studios that are 'looking for clues to box office

success.’ However the company acknowledges that this script analysis ‘is not done by machines.’ So this does not really qualify as a true data analytics activity as it relies on human intuition rather than machine algorithms to derive interesting correlations.

Dr. Peter Gloor at the Center for Collective Intelligence at MIT’s Sloan School of Management accurately predicted the 2007 Best Motion Picture of the Year Academy Award for *The Departed* (2006) by tracking:

...a year’s worth of Oscar conversation on IMDB, use regression analysis on its word counts, and plug that into a custom-designed index to predict Oscar nominees and winners.

(see <http://www.vanityfair.com/online/oscars/2013/02/mathematical-formula-oscar-winners-imdb>, accessed 08 March 2012).

Prediction markets like Intrade.com (suspended when accessed on 29 March 2013) and farsiteforecast.com are among a number of websites that have also engaged in Oscar predictions by tapping into ‘the wisdom of crowds’ via social networking sites. Effective crowdsourcing of big data in this way not only depends on a lot of data being generated by and publicly available from, the crowd, but also on the quality of the crowd generating the data, as farsiteforecast.com point out,

Essential to analyzing the success and value of crowd sourcing is a firm grasp of one thing – the crowd. The question is – who comprises the crowd

(see <http://www.farsiteforecast.com/vanity-fair-polls-and-parties/>, accessed 18 March 2013)

and emphasize in figure 1.7 below:

Crowd-Sourcing Your Way To The Stage

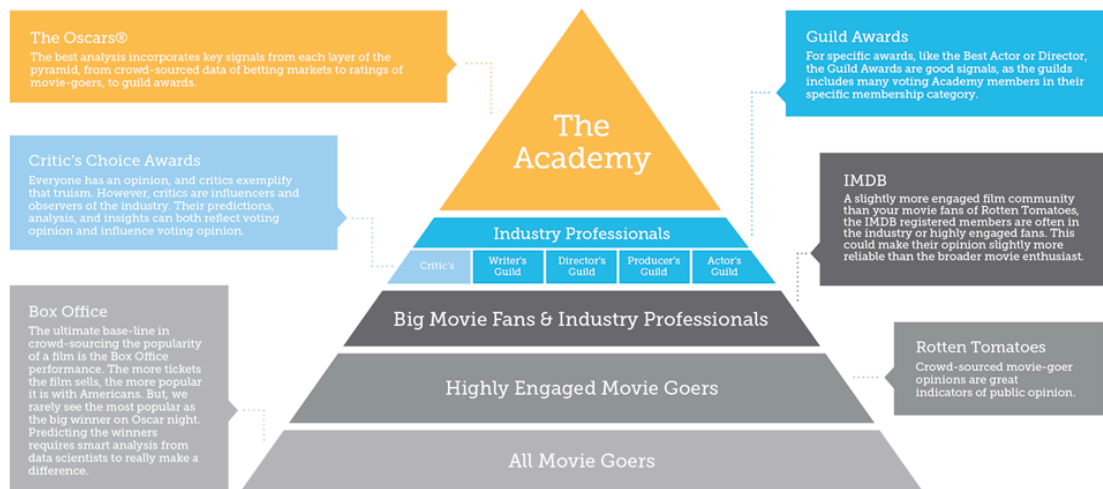


Figure 1.7 – Quality Pyramid of Crowdsourced Data

But Oscar predictions like this using social networking buzz to predict success are not based on analyzing screenplay content (predictive analysis of screenplay content is discussed below) however big data techniques are being used as a means of comparing individual screenplay content to relevant big data content as discussed in 'dialog analysis' below.

1.12 Screenplay APIs

Once you have datafied a screenplay it becomes practical and perhaps useful to permit interaction with that data by applications other than the 'authoring' application (i.e. the 'authoring' screenwriting tool). This is achieved by using an application programming interface or API designed to allow other applications to read data from the screenplay or write data to the screenplay.

A key reason why certain social networking sites like Facebook and Twitter and many other Web 2.0 sites generally have grown so rapidly and managed to build an application ecosystem around what may be a very simple concept (e.g. posting a 140 character Tweet) is by providing an Application

Programming Interface (API) to allow other applications to interact with them. The provision of an API enables the global developer community to get more value from or add value to the API source application.

The basis of any API can be summarized in two words 'GET' and 'PUT'. A GET retrieves data from an application and a PUT posts data to an application. So a GET could retrieve all Tweets posted to a user's account subject to specific criteria and a PUT could post a Tweet to that account from a different application than Twitter. Obviously APIs restrict what data can be retrieved from an application and what data can be posted to it but essentially they open up applications to the outside world through what is designed to be a well-defined and secure interface.

A screenplay could function as the 'object' of an API for example to get data from the screenplay text or to post data to it. But this is only realistic if the screenplay is datafied e.g. stored in an XML file or a database. If a corpus of screenplays were stored in an online archive in a standardized data format with an API this might make various kinds of analysis by different communities of interest- genre analysis, 'potential blockbuster' predictive analysis and sentiment analysis, not to mention general academic research – much easier to do.

A screenplay API might assist when writing is being done collaboratively with a partner or group or as a way for production managers to interact with a script during production of the movie. Access to an API is also a way to facilitate interaction with a screenplay as a data 'source/target' from devices like mobile phones, where perhaps using the main screenwriting application does not make sense. So if you want to post content to the screenplay, captured on the device (e.g. a photo taken on location or an idea or note scribbled on the spur of the moment) then an API would help to enable you to post this data from your phone app to the screenplay data repository.

1.13 Screenplay as Data Wall or Stream

Online social networks Facebook and Twitter have popularized the concept of data-walls used to display data-streams. Users 'poke' data onto their personal Facebook wall, which may then be pushed on to the walls of their friends. Users 'tweet' data into their personal Twitter stream, which may then be 'retweeted' into the streams of their followers. And there is no reason why a screenplay cannot be viewed in the same way both in terms of its core content and enrichment content.

Here, for example, core content such as scenes and dialog/action data could be viewed as being 'poked' onto the screenplay as data wall. And enrichment content such as comments about a scene could be viewed as being 'tweeted' to the scenes data stream. Online social applications also tend to make user activity more transparent, by displaying all user interaction in near real-time in the form of an activity stream, and again this is equally applicable to a screenplay in order to make more transparent and track the contribution made to the dynamic screenplay artefact by all the stakeholders in the community of interest around the script.

Indeed this activity stream tracking could even have a commercial dimension in terms of calculating whether or not a specific stakeholder can claim a writing credit on the script based on their content contribution, if the script is in fact made into a movie.

1.14 Screenplay as Interface

Again, the datafication of the screenplay changes the way that an individual user may interface with the screenplay. The prevailing interface paradigm is to present the screenplay through the interface of 'script' – the vertical scrolling 'toilet paper' paradigm of blog posts and comments – or through the interface of scene - a sheet or two off the roll. But with screenplay as data there is no longer a restriction on the way a user interfaces with a screenplay.

For example:

- A writer may want to interface with the script content by working only with scenes that have a status of 'draft' versus 'final'; or with scenes that are specifically shared with her by a partner; or with scenes that have recently attracted the most comments from others with access to the script.
- An actor may want to interface with the script by working only with scenes that she is in and then maybe toggling the scene content so that only her dialog is shown; or viewing all her dialog only across the whole script.

This is much easier to do if the screenplay content is stored as data. Suddenly the traditional whole-script or even scene-based content 'straight-jacket' can be discarded and users allowed to interact with the screenplay content through an interface that reflects their role or what it is they actually want to use the content for. The screenplay as data enables the user to choose or personalize their interface with the content.

Summary

It is clear that if we consider screenplays as data-centric rather than document centric artefacts that comprise both data and metadata, we get a different perspective of screenplay content and how it can be organized and analyzed. Plain text or and other unstructured data storage formats do not provide the same rich potential for analysis as screenplays stored in tabulated (SQL database), documented (document database) or tagged (XML) structured data storage formats. Datafied screenplays are an essential foundation for 'what happens next'.

Screenplays stored as data are also essential to enable the application of big data techniques to analyze a corpus of screenplays and to 'open up' screenplays so that a range of different applications can interact with the content via a screenplay API. Although there is not yet a standard screenplay

format used by all screenplay applications, as say XBRL is used for mandatory corporate reporting in many countries, the *Final Draft .fdx* format is a proto-standard that could be adopted industry wide by screenwriting applications in the future.

In chapter 2, I discuss how screenplay as data provides the foundation for screenplay analytics – the analysis and visualization of screenplay content - helping readers to better understand a script based on the evidence of the data and helping writers to ask questions of their script by surfacing both 'knowns' and 'unknowns' from the evidence of the screenplay data.

2 Screenplay Analytics

Many screenwriting applications do not focus much or any attention on content analytics, thus depriving the screenwriter of potentially valuable tools for asking more questions of their script in order to improve its quality and/or commercial potential.

Here, screenplay analytics means applying analytic techniques and algorithms to a screenplay text – both metadata and data - using a software tool. Analytic techniques and software can be used for various purposes and come in various forms. Analysis can be used for specific purposes such as pattern recognition or to summarize and aggregate data so it can be visualised. The insight derived from this analysis can be used for decision support to facilitate risk analysis or fraud detection for example. Analysis may be delivered in the form of row/column reports, graphs and charts or visual infographics, whatever makes it easier to assimilate and understand the information in order to facilitate decision-making. For some time, in the world of business software applications, ‘analytics’ was more often referred to as ‘decision support’ as this is often what it is used for.

In the domain of theatre, academic linguistic analysis of the works of Shakespeare – both as individual works and as a corpus – extends back at least to Caroline Spurgeon’s 1935 work, *Shakespeare’s Imagery and What it Tells Us* and includes studies by Mahood (1957), Carroll (1967), Elam (1984) et al. as well as Shakespearian dictionaries from Schmidt (1902) to Crystal (2002) and concordances from Beckett (1787) to Spevack (1968-70). However these linguistic analyses have focused on the use of language, the constituent words and the identification of thematic patterns within a very specific content domain – that of the works of Shakespeare - rather than on the development of analytical models and visualizations that can be applied to these and other non-theatrical dramatic works like screenplays with equal utility.

My focus here is less on linguistic analysis of the language of an individual screenplay or that of a corpus of screenplays, say within a specific genre, but on the delivery of basic screenplay analysis and visualization tools to help the various stakeholders in the screenplay, to analyze and visualize the content of a script in various ways, including its use of language. In the context of single feature film screenplay, we are interested in analytics that can positively influence the restructuring or rewriting of the screenplay content not only to effect a quantitative (i.e. to cut or add content) impact on the content but also to facilitate a qualitative (i.e. to revise content) improvement. The current generation of screenwriting software does a poor job of helping screenwriters to restructure or rewrite their scripts because so little in the way analytic functionality is supported

That the process of script rewriting, rather than script writing, is so poorly supported in the current generation of screenwriting applications is surprising because it is well established that screenwriting is a process and that screenplays typically go through any number of rewrites during development and while they are being shot in production. According to Leily Kleinbard in *The Atlantic* (Nov. 21, 2012), David Magee, the screenwriter of Ang Lee's *Life of Pi* (2012), originally considered Yann Martel's book 'unfilmable', which may be why adapting the book into a screenplay took '170 Script Revisions'. Perhaps some kind of screenplay content analytics could have cut the number of revisions needed and therefore reduced the script development cost?

Screenplay analytics is likely to be of most benefit not just to writers but also to readers of scripts, particularly the story analysts, story editors and 'polishers' employed by studios or production companies to evaluate or improve the 'quality' of a script. This is because analytics is a way to learn more about a script prior to reading it or to confirm or deny intuitive conclusions based on the evidence of the data after it is read.

According to Garfinkel in *Screenplay Story Analysis* (2007:xv), for an experienced screenplay reader/analyst, 'it takes an individual about one

and half hours to read a decent, readable screenplay’.

However, it would take several more hours, even for a professional script reader, to gain a reasonable understanding and appreciation of the quality of the script and to evaluate the commercial or artistic potential of the screenplay. This is precious time for production company executives and studio script readers who apparently have a never-ending “slush-pile” of speculative scripts scheduled for reading and appraisal or that are submitted online to various script ‘harvesting’ web sites that have sprung up to encourage and manage this stream of ‘user generated’ content.

According to Edwards and Skerbelis (2009:22-25), major Hollywood studios tend to have 8-12 readers/analysts on staff who are each reading 8-10 scripts a week to produce a range of coverage and other content analysis reports and it is ‘the story editor’s responsibility to evaluate readers’ coverage on a regular basis and to maintain the quality of that coverage.’ So anything that can help to cut the time to do the evaluations or improve the quality of the results either by analysis of an individual screenplay or as a result of analysis of data-based results over time, could enable studios to read more scripts (increase throughput) or make better decisions (improve quality).

The lack of attention paid to screenplay content analytics (as opposed to say movie analysis and Oscar predictions) by both theorists and practitioners is also surprising because analysis and visualization of the dramatic form is not a new concept. Aristotle’s *Poetics* is widely regarded as the earliest work addressing aspects of the analysis of drama and this work even includes a reference to visualization (1995:27), as Aristotle recommends that,

*“When constructing plots and working them out complete with their linguistic expression, one should as far as possible visualize what is happening. By envisaging things very vividly in this way, as if one were actually present at the events themselves, **one can find out what is appropriate and inconsistencies are least likely to be overlooked.**”* (my emphasis).

Clearly Aristotle was not referring to the analysis and visualization of screenplay content, but his recommendation is relevant here. Probably the first published example combining both the analysis and visualization of a dramatic form is Freytag's pyramid (figure 2.1), from his *Technique of the Drama* (1899:115 - originally written in 1863) that visualized his definition of the five parts of the drama as a simple pyramid - introduction (a), rise (b), climax (c), return or fall (d) and catastrophe (e).

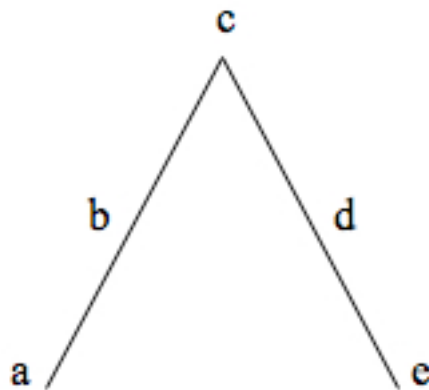


Figure 2.1 - Freytag's Pyramid

Things have changed a lot since Freytag's day, especially in the last few decades, as software has emerged that can automate many content analysis and visualization tasks (although we should be clear that you cannot find a screenwriting tool that can generate a Freytag's pyramid for you). For example, software can generate visualizations from the content of specific Shakespeare plays, such as the social network of *Hamlet* (see figure 2.2 below from

<http://services.alphaworks.ibm.com/manyeyes/view/SjjqGFsOtha6ymEZWajPF2>, accessed 23 September, 2008), and to provide a completely new interface for viewing and interacting with the text of a dramatic work, as exemplified by *Watching the Script* (see <http://digitalplaybook.humviz.org>, accessed 23 September 2008) – a content navigation interface shown in figure 2.3 below. The software used for both of these examples could be

applied equally well to screenplay analysis and visualization.

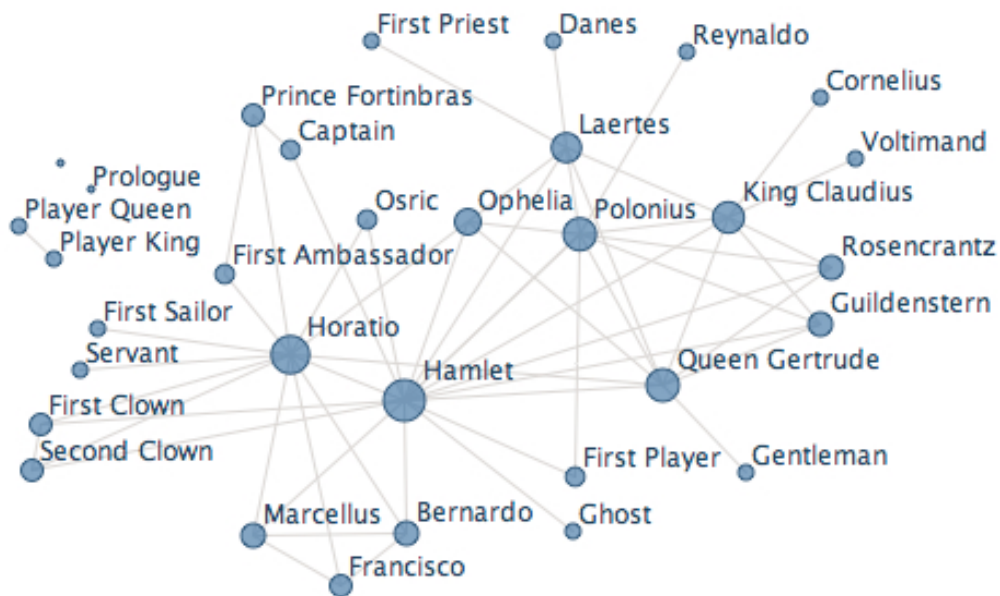


Figure 2.2: Character Associations in Shakespeare’s Hamlet

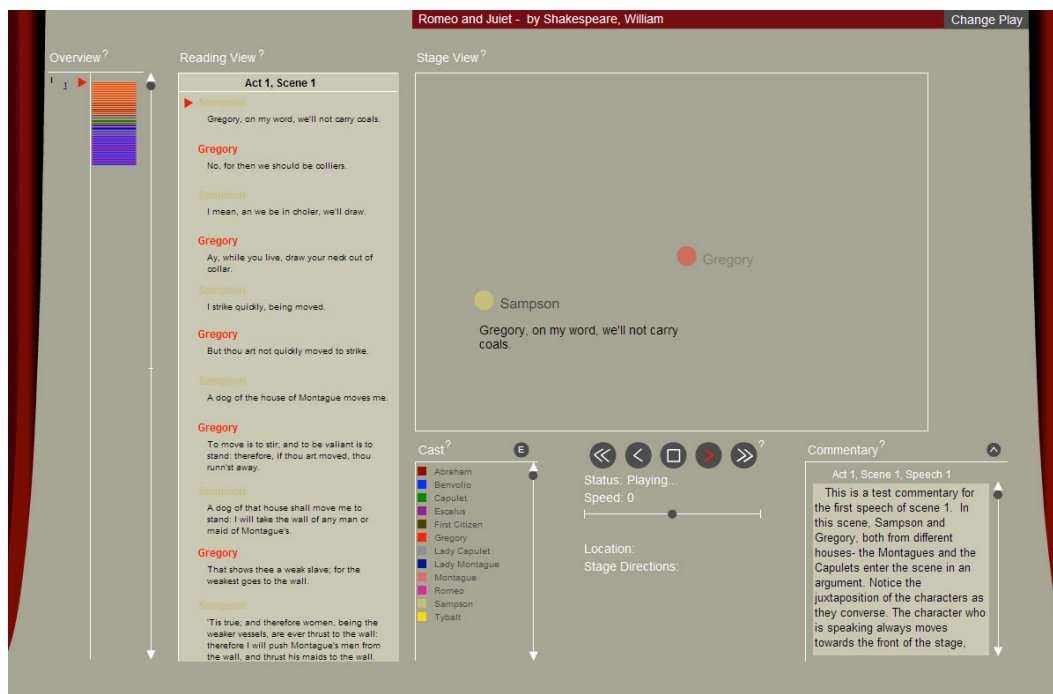


Figure 2.3 - Watching the Script

At the heart of what screenplay analytics is about are Corrigan's (2007:22) comment 'Be prepared with a questioning mind from the beginning' and Kenning's (2006:15) 'To diagnose your script's problems, you will ask questions of your script – a lot of questions'. Also, according to Card, Mackinlay and Shneiderman (1996), the purpose of visualization is 'to aid analysis and facilitate discovery, decision-making and explanation.' So my proposition is that by providing the screenplay as data it will be possible to deliver a range of analytical reports and visual images as output, to enable any screenplay stakeholder to ask more questions of the script and hopefully to make more informed decisions about it.

The intention of this questioning process is primarily to help the writing stakeholders (i.e. the writer and/or co-writers) in particular to better understand where to focus their attention in order to raise the quality of their screenplay and perhaps increase the salability or competitiveness of their script. There is certainly a commercial dimension to screenplay evaluation that is relevant to screenplay analytics and visualization. In *Writing Drama* Lavandier states this most forcefully,

I am convinced that the health of the film (and television drama) industry hinges on the ability of all those involved in decision-making to read, **that is to evaluate**, a screenplay correctly. (my emphasis). (2004: 26)

Here it is important to clarify that by visualization, we are primarily concerned with visualizing the screenplay content (input) rather than visualizing the possible filmic (output) from the screenplay. Arguably, a screenplay is itself a form of narrative visualization since it enforces a specific presentation 'form' on the content. Yet the screenplay is not one of the 7 genres of narrative visualization proposed by Segel and Heer (2010), presumably because it is not 'visual' enough compared to their other genre selections, despite the fact that even the traditional paper-based screenplay format is itself a visualization of the data and metadata of a script, albeit a simple one.

Visualizing the output of a screenplay is the realm of artistic storyboarding and pre-visualization software and the movie production process itself. In this context, both storyboarding and pre-visualization – when initially performed in an automated fashion by software – are an existing form of screenplay analytics that is well-supported in current tools. Here the screenplay content becomes the input that is analyzed to generate the basis for a storyboard or pre-viz that can be further enhanced by visual artists using the software. The analysis consists of identifying scenes in the screenplay and visual entities such as characters and objects for example so that these can be matched to visual assets in the software’s database and selectively rendered as ‘proto’ storyboards or pre viz animations.

Storyboarding involves manually or electronically (i.e. using a storyboarding software package) sketching out a visual impression of the individual camera shots required to shoot a specific scene or set of scenes in the movie described by the screenplay. The storyboard is a collection of "boards" or individual drawings, in shot sequence, that give an idea of the setting and props, the character action, the prop or camera movement, the point of view and the general look and feel of the scene. The primary purpose of a storyboard is to help the director and/or his production managers imagine, block and plan the shot.

Using the screenplay as a data source to visualize an animated set of shots is known as “pre-visualization” or “pre-viz”. Pre-visualization describes the automatic creation of a visual product from the script by a software package, prior to shooting the movie with a camera. A pre-visualization is either static – in the form of computer-generated storyboard of static images - or animated in the form of some kind of computer-generated animation that can be ‘played’ on-screen. Like storyboarding, the primary purpose of a pre-viz is also to help the director and/or the production managers of a movie to imagine, block and plan the shots in a scene.

An animated pre-visualization refers to the use of animation software to provide some kind of basic ‘moving picture’ using the images suggested by

the screenplay text. By creating an animation involving images of the scene setting, the scene characters and their movements and perhaps connecting this to a synchronized audio track of the dialog being spoken and a backing soundtrack, a far more sophisticated level of pre-production visualization of the movie is possible.

Cavazza, Friedman, Liu, Ramirez and others have already done much work in the area of screenplay pre-visualization by applying Natural Language Processing (NLP) techniques and tools to screenplay texts. The results of this work include prototype tools such as *WordsEye* (Coyne and Sproat), *ScriptViz* (Zhi-Qiang Liu) and *Scenemaker* (Hanser, McKeivitt, Lunney and Condell), designed to help with the automated conversion of screenplay text into computer-generated animations. These animations typically include generating 3D characters and modeling their expressions and actions and visualizing background scene environments for the characters to be situated in. This in turn depends on the ability to identify emotions in the text, as expressed by the both the dialog and action of the characters, to significantly enhance the basic positive/negative evaluation delivered by sentiment analysis for example. This has been the focus of work by Strapparava and Mihalcea, and Aman and Szpakowicz among others.

Commercially released storyboarding and pre-visualization software such as *Frameforge 3D* and *Avid* for example, can already import a screenplay file generated by a compatible screenwriting package and automatically generate “skeleton” shot storyboards and/or basic scene animations directly from the script text.

Here we are concerned with ways of analyzing and visualizing the textual content of the screenplay primarily for the benefit of the screenwriter/reader, rather than to help the director or production managers imagine and plan shots or scenes in pseudo-production mode to create a “proto-movie”. The aim is to deliver a level of qualitative improvement prior to investing money and resources into any kind of pre-visualization, such as storyboarding or animation, both of which may be

relatively costly and time-consuming exercises in of themselves and in any case may be beyond the skills or interest of the screenplay author(s).

2.1 Why bother analyzing screenplay content?

Franzosi (2010:4-5) asks the question: 'Words are beautiful: Why do you want to turn them into numbers?' He answers his question thus: 'I quantify simply because I have far too much information to deal with qualitatively'.

Analyzing screenplays is not a well-established niche of analytics but analyzing released movies is fast becoming a popular pastime of bloggers, screenwriting manual writers and websites focused on the more sensational aspects of the movie world such as the Oscar awards. The web sites thecredits.org and brandwatch.com collaborated to analyze and visualize the social media 'buzz' around the 85th Academy Awards (2013) to try to predict the Oscar awards by using a social analytic algorithm (authored by Edward Crook) to analyze postings on public forums, Facebook, Twitter etc. (<http://www.brandwatch.com/2013/02/introducing-the-social-oscar-tracking-award-buzz/> - accessed 25 February 2013).



Figure 2.4 - Brandwatch Social Oscars DataViz

Brandwatch then produced a comprehensive ‘wisdom of crowds’ pre-Oscar infographic based on this social buzz, which proved remarkably accurate in terms of its Oscar award predications (see <http://www.brandwatch.com/2013/02/the-power-of-crowds-predicting-the-oscar-winners/> - accessed 25 February 2013). But these visualizations tap into a relatively big social dataset that reflects international interest in movies in general and the Oscars in particular. Screenplays do not generate the same general interest as movies and either they do not involve a social network at all or do not involve a significant enough ‘crowd’ from whom ‘wisdom’ can be sourced for analysis and visualization purposes.

Nevertheless, the content of a screenplay must be – and is - of interest to someone. A screenplay is part of a commercial value chain and, until it is realized as a movie, represents latent value for every stakeholder with a financial interest in the production and commercial success of the output movie. These stakeholders typically include the producer, director, key talent and the writer(s).

The writer is usually the initial beneficiary of this latent value. So what is a script worth to a writer? The Writer’s Guild of America (WGA) is certainly the largest and most influential body in the world that represents professional screenwriters. The WGA Schedule of Minimums (2004), effective for the period of 11/1/06-10/31/07, requires that WGA represented writers be paid between \$56,500 - \$106,070 for an original screenplay including treatment.

In reality, many “hot” screenplays are bought for considerably more than this WGA minimum. Using sources such as *Daily Variety*, *Hollywood Reporter*, DoneDealPro.com and Trackingboard.com, Scott Myers reported scripts selling for up to \$2 million on his blog (<http://www.gointothestory.com/2009/01/spec-script-sales-analysis-2008-big.html> - accessed 27 July 2009). However, very few spec scripts are

actually sold each year. According to *Hollywood Reporter* (Dec 23-25, 2006 issue), only 77 spec scripts were purchased in 2005 (<http://fencingwiththefog.blogspot.ca/2006/01/screenplay-sales-statistics.html> - accessed 18 March 2014).

Not only are screenplays worth considerable sums of money to a writer in upfront fees, there may be significant “back-end” compensation as well, if the writer’s contract includes rewrite fees or percentage points of the movie’s eventual gross profit. In this case the royalty revenue stream from the “points”, resulting from the initial release revenues and other residuals from DVD/cable/pay-TV release etc., may well-outstrip the upfront payment. In rough terms, the initial payment may equate to at least a reasonable year’s salary and the points royalties anything up to moderate personal wealth.

A screenplay is a valuable commodity to the selling screenwriter but can cost far more significant sums for the buyer to develop. In *The Hollywood Economist*, Epstein (2012:63) reports that the budget line for ‘Story and Rights’ for *Terminator 3: Rise of the Machines* (2003, Jonathan Mostow) was \$19.6 million, which presumably included the \$5.2 million he claims was spent developing the initial script (2012: 51). Although movie budgets, as reported by studios, are notoriously difficult to validate and verify.

A screenplay is an even more valuable commodity in an industry that makes significant financial bet on every screenplay given the green-light to proceed into production. We can get some idea of how valuable from these estimates provided by the Hollywood tracking website *The Numbers* (<http://www.the-numbers.com> - accessed 27 July 2009).

Until *Avatar* (2009, James Cameron), the highest grossing movie ever was *Titanic* (1997, James Cameron) with an estimated worldwide gross of \$1.849 billion against an estimated budget of \$200 million. The highest grossing movie of 2008 was thought to be *The Dark Knight* (2008, Christopher Nolan) with an estimated worldwide gross of \$1 billion against an estimated

budget of \$185 million. However Box Office Mojo claims that Avatar has now become the world's highest grossing movie, having grossed \$2.075 billion worldwide so far. (<http://boxofficemojo.com/news/?id=2667> - accessed 30 August 2012).

Clearly giving the green light to produce a movie from a screenplay is massively risky and has the potential to return either stellar profits or generate losses that could threaten to bankrupt studios or make them a takeover target (cf. United Artists and Michael Cimino's *Heaven's Gate* (1980)). That's why perhaps the only well-established branch of screenplay analytics is focused on the use of predictive analytics to use pre-defined criteria to try to predict a box office hit to help potential screenplay investors to minimize their risk.

The idea of defining rules to be applied to screenplays for evaluating the quality of the content is not new. One of the first sets of rules published specifically to guide screenwriters can be found in chapter XI of the *Palmer Plan Handbook* (1919). William C. DeMille's rules comprise ten qualitative rules governing 'Story Requirements' and a further ten qualitative rules to avoid screenplay rejection. A few decades later, Vale (1944:282-287) identified 142 'of the most common mistakes from which the questions necessary for analysis can easily be derived'. Rules can be found in many screenplay writing manuals such as Flinn's *How Not to Write a Screenplay: 101 Common Mistakes Most Screenwriters Make* and frequently popup on the web, for example the "22 Rules" proposed by Pixar storyboard artist Emma Coats (see <http://slacktory.com/2012/07/pixar-story-rules-illustrated-by-icanlegothat/> - accessed 3 September 2012).

However, it should be noted that no screenwriting applications today actually codify and apply any of these kinds of rules as a means to evaluate screenplay quality.

Predictive Analytics

Siegel (2013:3) makes the point that data ‘embodies a priceless collection of experience from which to learn’ and predictive analytics is all about analyzing data to make predictions based primarily on discerning patterns in the evidential data, often based on the application of domain-specific rules to test the data.

ScriptReader (<http://scriptreaderonline.com> - accessed 3 September 2012) provides a detailed checklist of questions/rules that claims to provide ‘a quantitative approach to screenplay analysis’ based on a proprietary ‘evaluation algorithm’ developed as a ‘software analysis tool for a major Entertainment Agency’ so writers can find out ‘if your screenplay will get a Pass, Consider or Recommend’. However this is a manual process that is in effect a self-assessment by the writer, which is then scored to indicate the “grade” the screenplay could get from an industry reader, rather than an automated process whereby the screenplay content is subjected to software algorithms that ask and answer these kinds of grading questions by generating analysis reports and visualizations at the click of a button.

However, according to *The Guardian* (July 13, 2007:3), a British firm called Epagogix says it has ‘designed a computer programme to assess a proposed movie’s likelihood of success...’ and ‘claims it can estimate 80% of projects’ likely US box office to within \$10m of the final figure.’ Apparently it does this by breaking down a script into ‘hundreds of constituent elements...assigning each one a commercial value.’ Over time, if this process results in high levels of predicative success, then the element mix and value assignments will be further optimized, as the program (or its designers) learn more about what makes a movie a “surefire” commercial success.

However the target market for the Epagogix commercial valuation service is not the screenwriter but the studio or production company. The sort of fees that studios might be prepared to pay for this kind of information can only be guessed at. In *The New York Times* article mentioned above relating to the

Worldwide Motion Picture Group, the amount paid for script analysis services could be 'as much as \$20,000 per script'. On this basis, the Epagogix business model is unlikely to accommodate individual screenwriters wishing to gain a better understanding of the commercial potential of their screenplay. Yet the fact that Epagogix, Worldwide Motion Picture Group and others exist at all and studios are apparently willing to pay for their services is an indication that predictive screenplay analysis is a viable and valuable service proposition.

Epagogix and other service providers – such as Dave Kelly Entertainment, which claims to use a package called Gold Light to perform proprietary analytics on screenplays (See <http://www.davekellyentertainment.com/marketsolution.html> - accessed 11 August 2007) - are almost certainly leveraging software tools and techniques known collectively as “predictive analytics”.

Eliashberg, Hui and Zhang (2006) discuss the use of predictive analytics as a means to determine a movie's potential success. In an appendix (2006:27), the authors propose 22 questions that can be used to help gauge the potential success of a movie based on the characteristics of the script. These questions are interesting in the context of this thesis so they are reproduced below in full:

- 1) *Clear Premise (CLRPREM)*: The story has a clear premise that is important to audiences.
- 2) *Familiar Setting (FAMSET)*: The setting of the story is familiar to you.
- 3) *Early Exposition (EAREXP)*: Information about characters comes very early in the story.
- 4) *Coincidence Avoidance (COAVOID)*: Story follows a logical, causal relationship. Coincidences are avoided.
- 5) *Inter-Connected (INTCON)*: Each scene description advances the plot and is closely connected to the central conflict.

6) *Surprise (SURP)*: The story contains elements of surprise, but is logical within context and within its own rules.

7) *Anticipation (ANTICI)*: Keep readers trying to anticipate what would happen next.

8) *Flashback Avoidance (FLHAVOID)*: The story does not contain flashback sequences.

9) *Linear Timeline (LINTIME)*: The story unfolds in chronological order.

10) *Clear Motivation (CLRMOT)*: The hero of the story has a clear outer motivation (what he/she wants to achieve by the end of the movie).

11) *Multi-dimensional Hero (MULDIM)*: Many dimensions of the hero are explored.

12) *Strong Nemesis (STRNEM)*: There is a strong nemesis in the story.

13) *Sympathetic Hero (SYMHERO)*: Hero attracts your sympathy because he/she exhibits courage AND belongs to one of the followings: -good/nice, funny, good at what he does OR has power.

14) *Logical Characters (LOGIC)*: Actions of main characters are logical considering their characteristics. They sometimes hold surprises but are believable.

15) *Character Growth (CHARGROW)*: Conflict is important enough to change the hero.

16) *Important Conflict (IMP)*: The story has a very clear conflict, which involves high emotional stakes

17) *Multi-Dimensional Conflict (MULCONF)*: The central conflict is explained in many different points of view.

18) *Conflict Build-up (BUILD)*: The hero faces a series of hurdles. Each successive hurdle is greater and more provocative than the previous ones.

19) *Conflict Lock-in (LOCKIN)*: The hero is locked into the conflict very early in the movie.

20) *Unambiguous Resolution (RESOLUT)*: Conflicts is unambiguously resolved through confrontation between the hero and nemesis at the end.

21) *Logical Ending (LOGICEND)*: The ending is logical and believable.

22) *Surprise Ending (SURPEND)*: The ending carries surprise and is unexpected.

In the opinion of Eliashberg et al. positive answers to these questions are likely to indicate good potential for commercial success, therefore purchasers of screenplays may have a clear interest in analyzing screenplays by leveraging this methodology.

In their 2010 paper, *Green-lighting Movie Scripts: Revenue Forecasting and Risk Management*, Eliashberg, Hui and Zhang state that:

As a script is the very foundation of a movie, a sophisticated analysis of the textual information and hidden story structures in the script help us better predict box office revenues.” (2010:3)

By analyzing the genre/content, words and semantics of a script and using them as predictors in a Bayesian Additive Regression Tree for Quasi-Linear model (BART-QL), the authors’ claim that:

...our model’s capability to generate predictive distribution of the box office revenue not only allows a studio to assess the risk associated with a point forecast, but also opens new doors for a studio to optimize its portfolio choice and manage its risk exposures.

- a model one can imagine that every production studio would want to apply before investing further funds in script development.

Sentiment Analytics

Other than predictive analytics, another kind of analytics that could be applied for screenplay analytics is sentiment analysis. Sentiment analysis is used to deduce positive, neutral and negative sentiment from a body of text. It has become a popular way to indicate the sentiment of the ‘crowd’ – for example to determine what is trending on Twitter - by analyzing data streams on social networks. For example, it has been used to analyze sentiment about products from tweets posted to Twitter (see Go, Bhayani and Huang at

<http://cs.stanford.edu/people/alecmgo/papers/TwitterDistantSupervision09.pdf> - accessed 11 June 2013) and to analyze responses to movies via reviews posted to Digg (see Yessenov and Misailovic at <http://people.csail.mit.edu/kuat/courses/6.863/report.pdf> - accessed 11 June 2013) and via movie reviews posted on discussion boards (see Thet, Na and Khoo at <http://jis.sagepub.com/content/36/6/823.full.pdf+html> - accessed 11 June 2013).

Sentiment analytics can be applied to the text of an individual screenplay, rather than a data stream, in order to indicate its overall 'positivity' or 'negativity' at the script, sequence/storyline or scene levels. It can be used for a number of purposes:

- To identify positive or negative action or dialog snippets in order to help the writer to increase/decrease this positivity/negativity in the script
- To confirm, from the data, a tough start to a script or a happy ending.
- To differentiate conflict (negative) from resolution (positive).
- As the basis to visualize a positive/negative 'rollercoaster' chart from the script (see figures 2.5a/b below).

Of course it could also be used to determine the sentiment 'heartbeat' characteristic of a corpus of screenplays within a genre in order to understand if a specific screenplay conforms to or subverts genre sentiment conventions.

The positive/negative aspect of sentiment analysis is reflected in some 'story chart' visualizations from James Dai and Robert McKee. In both cases, the visualizations are based on multiple plotlines within the screenplay and both also highlight structural plot points.

James Dai's Story Charting (see <http://www.storycharts.ca/pages/casablanca/> - accessed 11 June 2013)

combines some interesting structural propositions with useful visualizations of plot lines and turning points In his analysis of *Casablanca*, his Story Chart visualization (see figure 2.5a) charts three plotlines:

1. External plot – Escape Quest (Laszlo and Ilsa)
2. Relationship plot – Love between Rick and Ilsa
3. Internal plot – Rick’s patriotism

The circles represent the key turning points in those plotlines with a positive (above the line) or negative (below the line) impact over time.

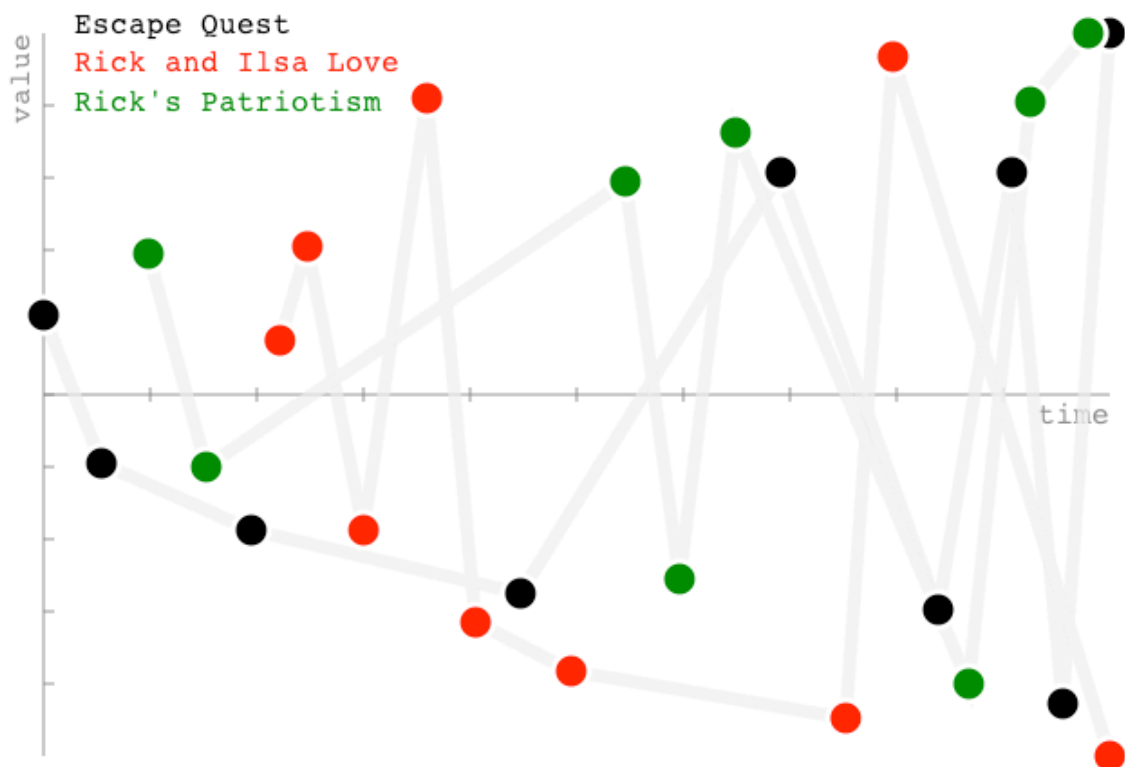


Figure 2.5a – Story Chart for Casablanca

Figure 2.5b is another kind of story chart, this time of Tarantino’s *Pulp Fiction*, and is attributed to Robert McKee (see <http://artscienceblog.blogspot.co.uk/2012/04/should-good-brand-stroy-chart-like-good.html> - accessed 11 June 2013). Here the positive and

negative 'planes' are superimposed on the acts and sequences and the inciting incidents, turning points and climaxes more clearly indicated along the plotlines.

Pulp Fiction

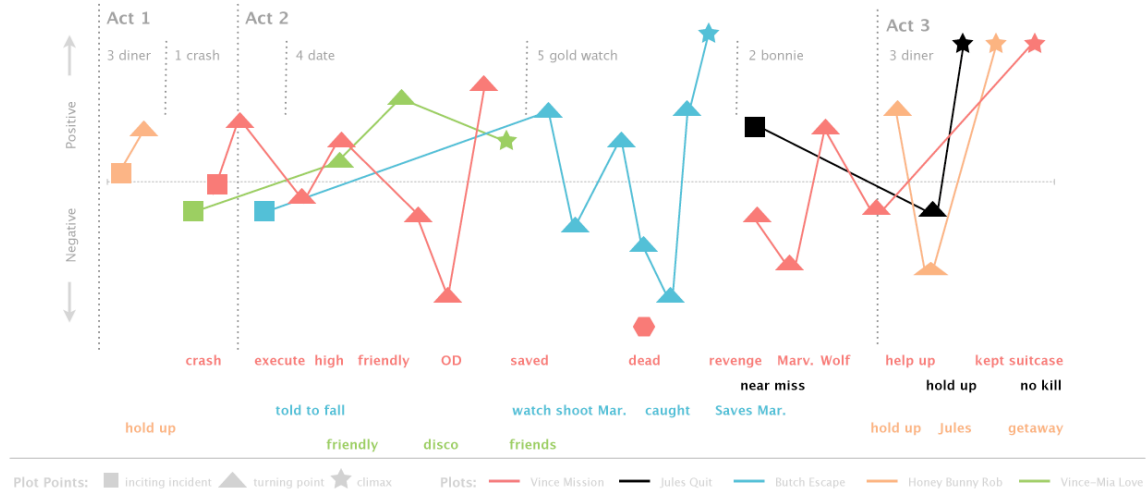


Figure 2.5b – McKee’s Story Chart of Pulp Fiction.

There are clearly viable commercial reasons for analyzing screenplay content and people already delivering it as a business offering. So what is worth analyzing?

2.2 What is Worth Analyzing?

Corrigan (2007:36-81), discussing analyzing and writing about film, suggests a number of topics for film analysis and writing that are equally applicable to screenplay analysis including: Themes, narrative, characters and point of view. A cursory review of the table of contents from almost any “how-to” screenwriting manual may also act as a guide to what is worth analyzing. For example, table 2.1 lists a representative range of topics from a random set of ten screenwriting manuals to show how some topics may be obvious targets for analysis and visualization (an X indicates that the topic is either explicitly mentioned in the contents or has an appropriate level of referencing in the index).

Screenwriting Manual	Character	Dialog	Genre	Narrative	Plot	Scene	Sequence	Structure	Theme
<i>Advanced Screenwriting</i> (Seger, 2003)	X	X	X	X		X		X	X
<i>(The) Art and Science of Screenwriting</i> (Parker, 1999)			X	X	X	X	X		X
<i>Crafty Screenwriting</i> (Epstein, 2002)	X	X	X		X				
<i>Screenplay</i> (Field, 2005)	X			X	X	X	X	X	
<i>Screenwriting Updated</i> (Aronson, 2001)	X								
<i>Story</i> (McKee, 1998)	X		X			X		X	
<i>Writing a Great Movie</i> (Kitchen, 2006)	X				X		X		X
<i>Writing Drama</i> (Lavandier, 2005)	X	X		X	X			X	
<i>Writing Movies</i> (Gotham Writers Workshop, 2006)	X	X			X	X			X
<i>Writing the Screenplay</i> (Armer, 1993)	X	X		X				X	

Table 2.1 - Potential Screenplay Analysis Topics

Based on this table, some obvious potential analytic subjects are:

- character (and dialog)
- genre (and theme)
- narrative (and plot)
- structure (including sequences and scenes)

So in order to narrow the scope of screenplay analysis and visualization discussed here, I will focus my attention on these two topics: Character and Narrative, because I believe that the interplay of character and narrative represent the essential DNA of a screenplay, they are easier for me to apply analytics to and do not require access to or the creation of a script corpus as genre analysis would do. My discussion of potential character analytics will embrace both action (what characters do) and dialog (what characters say) elements of the screenplay. My discussion of narrative will embrace plot, structure and theme.

I have decided not to include discussion of genre analysis for the following reasons. Firstly, there is no clear agreement about what the range of genres actually is or what exactly defines a specific genre. In other words there is no universally accepted standard to work from, for as Dancyer suggests in *The flexibility of genre* (in *Analysing the Screenplay*, 2011:122), 'Genres are not fixed forms. They are no more than templates for audience recognition and interest.' This is further reflected in De Haan's comment (2004:7) in *The Consulting Process as Drama* relating to client and consultant that could equally well apply to an audience and a movie perceived to be of a specific genre, 'Experience shows that usually the entry "starts before it starts": both the client and the consultant have images and expectations of the other, even before they meet.' And Edgar's reference to Terence Hawkes (2009:66) that "genre enables the reader 'to *decode* literature in the same mode as it was *encoded* by the writer'". This is as it should be because genre is a dynamic and developing notion that reflects the times, culture, audience whims and so on.

Secondly, I do not believe that screenwriters writing a spec script set out to "write to genre" - unless they specifically decide to do so or are specifically commissioned to do so and then the genre conventions that they pay attention to will be modeled on a specific exemplar movie or have been defined for them as specific instructions (e.g. "go write *Jaws* in space"). Thirdly I suggest that genre is largely a marketing decision that is either the driving genesis of the screenplay or grafted onto a screenplay later, triggering some kind of "adapt to genre" rewrite. Fourthly it is not entirely clear that any movie is necessarily representative of a single genre. For example, is Ridley Scott's *Alien* (1979) a representative of horror, sci-fi, feminist or even western genre or is it an example of some kind of composite genre? For all these reasons, I view screenplay genre analysis as probably the least useful to focus on, especially in terms of using analytics to improve the quality of an individual script.

However, this is not to say that genre analysis is not an interesting area for

screenplay analytics. Walker, Lin and Sawyer analyzed a corpus of 862 film scripts from the Internet Movie Script Database (IMSDb) in order to learn about and characterize character style from the dialog spoken in the scripts so that the results can be used as parameters to feed into Mairesse and Walker's PERSONAGE (personality generator) tool. Blackstock and Spitz also analyzed 399 scripts in order to determine they could successfully classify the genre of a script, Their techniques proved to be 'up to 55% accurate at nailing every single genre for a given movie exactly.' Genre analysis, as a niche of screenplay analytics, clearly has much potential for future research.

In each sub-section that follows, wherever possible, I will first outline the scope of analysis and visualization topic, then suggest:

- why a specific kind of analysis may be useful and benefit the screenwriter
- how the analytical subject in question may be identified within the screenplay content *programmatically*
- what form any visualization of the analysis may take

2.3 Character

Aristotle (1996:12) believed that character takes second place to plot and that people can be differentiated in only two ways – by defect or excellence (1996:5). Whereas in *The art of fiction* Henry James (1963:80) was more ambivalent, asserting: 'What is character but the determination of incident? What is incident but the illustration of character?' which sounds remarkably like George Eliot (from *Adam Bede* as quoted by Chatman 1978:96), 'Our deeds determine us, as much as we determine our deeds'. Culler (cited in Rimmon-Kenan 2002:31) offers the perspective that 'The notion of character, structuralists would say, is a myth.' Whether mythical or not, Rimmon-Kenan (2002:31) identifies two problems with characters: Are they about people or words or about being or doing? Or put another way, in screenplay as data terms, are they about dialog or action snippets?

Some popular screenwriting gurus regard effective characters as an essential part of every screenplay and critical to the quality and commercial potential of every script. In *Screenplay: The Foundations of Screenwriting* Field (2005:46) says, 'Character is the essential internal foundation of your screenplay. The cornerstone. It is the heart and soul and nervous system of your screenplay.' Corrigan (2007:44) states that characters '...normally focus the action and, often, the themes of a movie.' Lavandier (2005:110) believes that 'Drama consists of imitating human actions therefore of creating characters...' McKee's *Story* (1998:106) asserts that, 'The function of CHARACTER is to bring to the story the qualities of characterization necessary to convincingly act out choices.' In screenplay element terms, characters are usually involved in both a series of action (or scene description) blocks and in dialog blocks - either as messenger or receiver. This section will focus primarily on improving quality of characters in a screenplay by appropriate analysis of what a character does and says.

A movie without strong characters we believe in and can empathize with is almost certain to fail. This is because it is difficult to care about a movie as a whole, if there are no characters that we can relate to, that we can root for, that we can despise or love, or that we can identify with in some way. We want to be shocked or disgusted by characters, to be comforted or turned-on by characters, to be interested in what happens to them and why they are the way they are and do the things they do. Much of the rationale for audiences ignoring the fatigue of cinema movie-watching - concentrating on a darkened screen for around two hours - and staying to the end of a movie is tied up with finding out what happens to specific characters or how specific character development or relationships will play-out, or as Indick (2004: xi) puts it: 'Through the unconscious process of "**identification**," the people in the audience actually become the characters that they identify with in the script.'

There's no doubt that it's not only audiences that value characters, so too does the marketplace. As Elberse relates in *Blockbusters* (2013: pp.48-55)

characters that can drive blockbuster movies are worth a fortune. In 2009 Disney paid \$4 billion to purchase Marvel Entertainment (2013:48), a deal that has already resulted in billions of dollars of revenues from subsequent movie releases involving just a handful of Marvel characters (2013:55). Elberse (2013:51) quotes Bob Iger (Disney's CEO at the time) as saying 'This treasure trove of over 5000 characters offers Disney the ability to do what we do best' i.e. make money from movies. Assuming a few dud characters in the bunch it's fair to conclude that Disney paid around \$1 million for each Marvel character in the comic book publishers stable (including their linked merchandising capability).

However, what is it that defines characters in a screenplay? Is it only what Egri in *The Art of Dramatic Writing* calls the "bone structure" of physiology, sociology and psychology (2004:37-38)? Certainly one differentiator is their exterior characteristics – their physical and sartorial makeup. Another is their habits and "tics" – what Chatman (1978:127) called a set of "traits". But these obvious surface characteristics serve primarily as "identifiers" to differentiate characters visually for the audience, create some kind of expectation as to how these characters might behave as the movie progresses and use to remind the audience of a prior event (e.g. a facial scar from an important backstory confrontation).

Character is also more than the 'figural' aspects of characterization identified by Pfister in his *Theory and Analysis of Drama* (1991: 184). Pfister, in reference to a 'dramatic text', defines 'explicit' techniques as including 'self-commentary' and 'commentary by others' and 'implicit' techniques as non-verbal and verbal. If this were all a character comprised of in a screenplay, the character would likely be viewed in a stereotypical way that is unlikely to excite much interest or empathy from a movie audience.

But a more qualitative differentiator comprises their interior characteristics – their emotional depth, maturity and psychological profile for example. While it is possible that a character can change their physical appearance or traits over the course of a movie, it is more likely that changes in their

internal characteristics that will be of more interest to an audience, since these are “deep” changes in nature rather than “shallow” and more obvious surface changes. Just as it is more interesting to see how a business changes its internal culture than simply watch it move from one office block to another.

Another key aspect to character depth is how they relate to each other – the relational aspect of character. Imagine how much less interesting the Travis Bickle (Robert De Niro) character in Martin Scorsese’s *Taxi Driver* (1976) would be if all we ever saw and heard of him was as a lonely cab driver cruising the New York streets narrating his view of life and the city by constant voiceover. Bickle’s character becomes interesting as it is developed through his relationships with the two lead female roles of Iris (Jodie Foster) and Betsy (Cybill Shepherd).

Screenwriters may choose to use a character type or personality model to base their characters on. These include simple archetypes such as Indick’s 4-dimensional ‘Quatrains’ (2004:139) for female and male characters up to the particularly rich Enneagram model with its nine styles that Lee describes in *The Psychology of Screenwriting* as ‘a workable, accurate, and highly nuanced psychological tool for developing characters and understanding’ (2013:87). The Enneagram also benefits from the styles themselves being visualized, as shown in figure 2.6 below (see <http://www.9types.com/epd/1.php> - accessed 02 July 2013).

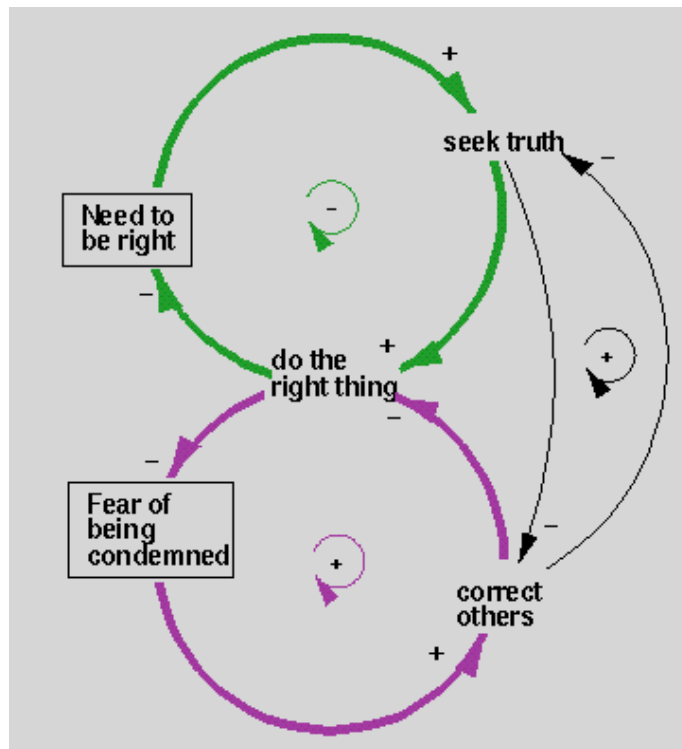


Figure 2.6 - The Reformer Enneagram Style

The fundamentals of screenplay character analysis (via technology) assume the following can be easily identified from the screenplay data:

- The character’s entrance (first appearance) and exit (last appearance) in the script.
- What dialog the character speaks and to whom.
- What action the character takes part in, where and with whom.
- Where the character is included in scenes that are part of the narrative line of specific plots/sub-plots/storylines in the screenplay.

The character’s entrance and exit is used to establish the extent of their “throughline” or participation in the script. Within this throughline will be blocks of dialog spoken by the character and blocks of action that the character participates in. Note that other characters may refer to a character

in dialog or action that takes place outside the actual extent of the character's physical throughline in the script. And in structural terms, characters participate in or are referred to in scenes. Individual scenes may form a component part of a plot or sub-plot thereby creating a plot-related (vs. a script related) throughline for the character also.

Role and Relationships

Characters in a screenplay may conform to a specific role dynamic and participate in a defined set of role relationships. The most obvious of these is the hero/villain or protagonist/antagonist roles and relationship. These roles and the effectiveness of this relationship are often considered to be central to the success of any screenplay. McKee (1998:379) claims that 'In essence the protagonist creates the rest of the cast' and visualizes the protagonist at the center of what he calls 'cast design'.

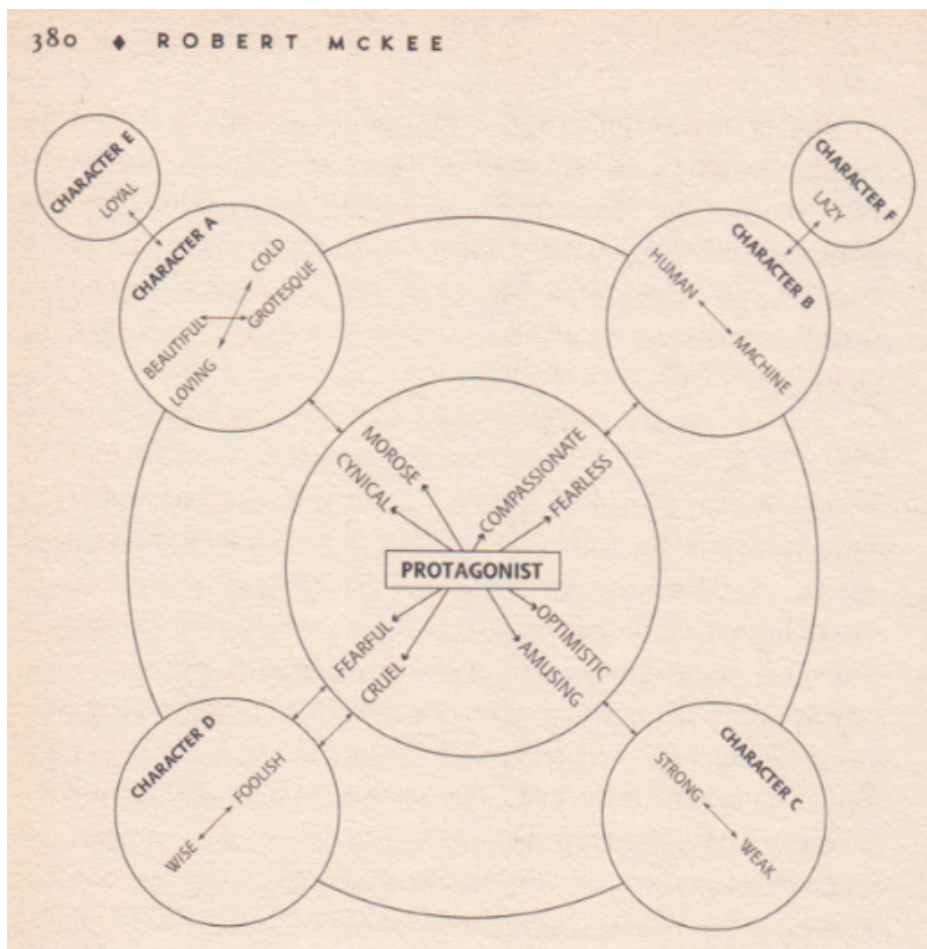


Figure 2.7 - Protagonist-Driven Cast Design (McKee 1998:380)

Baboulene's *The Story Book* emphasizes the story impact of the two key character roles through a basic Freudian analysis whereby the protagonist represents the 'super-ego' and the antagonist the 'id'. And it is the interplay of the two, in the form of conflict, which creates the resulting 'ego' or story: 'The true character of the protagonist and antagonist, revealed through the actions they take in response to the conflict between them.' [2010: 20]. In this sense, story is neither exclusively character nor plot, but requires characters and their actions to create and drive plot.

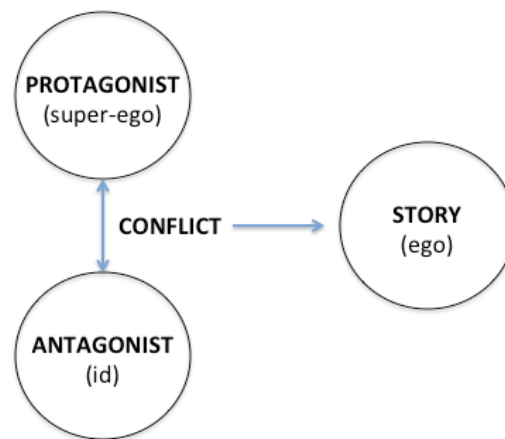


Figure 2.8 - Character conflict as story (adapted from Baboulene 2012:20)

Because of their relative importance in most screenplays (except say those with an ensemble cast), the protagonist/antagonist roles are worthy of particular attention for analysis. For example, who is the 'driver' - the protagonist or the antagonist? Is it the protagonist or the antagonist who drives the action by being proactive and making choices that 'pull' the other along with them? Are antagonist scenes triggered by a causal chain of preceding protagonist scenes or vice versa? Character analysis may be able to help a writer to better understand who really is the main character in their script and therefore who this script is really about.

But there are many other roles and relationships that add colour or depth to the central roles/relationships. These include ally/adversary roles, mentor

and shapeshifter roles, love-interest and sibling roles - among those roles identified in Vogler's *Hero's Journey* paradigm. These roles and relationships are often more interesting if certain "twists" are applied, for example if one is unknown to the other, if they are siblings at odds with one another or if one role acts as the "shadow" of the other (e.g. doppelgänger or 'mirror' roles - for example the Richard Gere and Andy Garcia roles in *Internal Affairs* (Mike Figgis, 1990) or the Ethan (John Wayne) and Scar (Henry Brandon) characters in *The Searchers* (John Ford, 1956).

Identifying

An indicator of role importance or at least "richness" could be how a specific role relates to others - either quantitatively or qualitatively - or put another way, the scope and depth of the social network of the character. A wider, deeper social network may indicate a more complex or sophisticated character. Here "wider" means interacting with more other characters and "deeper" means more interactions either in dialog or action. A wider, deeper social network also has a commercial dimension because these are the kinds of challenging roles that major acting talent is likely to be more interested in playing so a character analysis visualization could be used to help to 'sell' the role to an actor.

Visualizing

One way to analyze and visualize this aspect of a character in a screenplay is by using a node-based social network aka a friendship graph using clique detection algorithms. The user-selected "focal" character is centralized (see Hamlet in figure 2.2 above and 2.9 below (see <http://www.jibble.org/shakespeare/images/hamlet.xml-00000384.png>, accessed 6 July 2008) and the nodes that radiate from the focal character represent relationships with other characters. The relative size of each character node reflects the size of their action and or dialog elements in the script (i.e. bigger means more). The thickness of the connecting lines between characters reflects how often they interact together i.e. involved in

scenes or action or dialog together (i.e. thicker means more). The colour of the nodes reflects the “group” the character belongs to (e.g. protagonist and allies, antagonist and allies or “unaffiliated”). Social network visualizations would usually be whole-script based but could also be role, scene or storyline based for example.

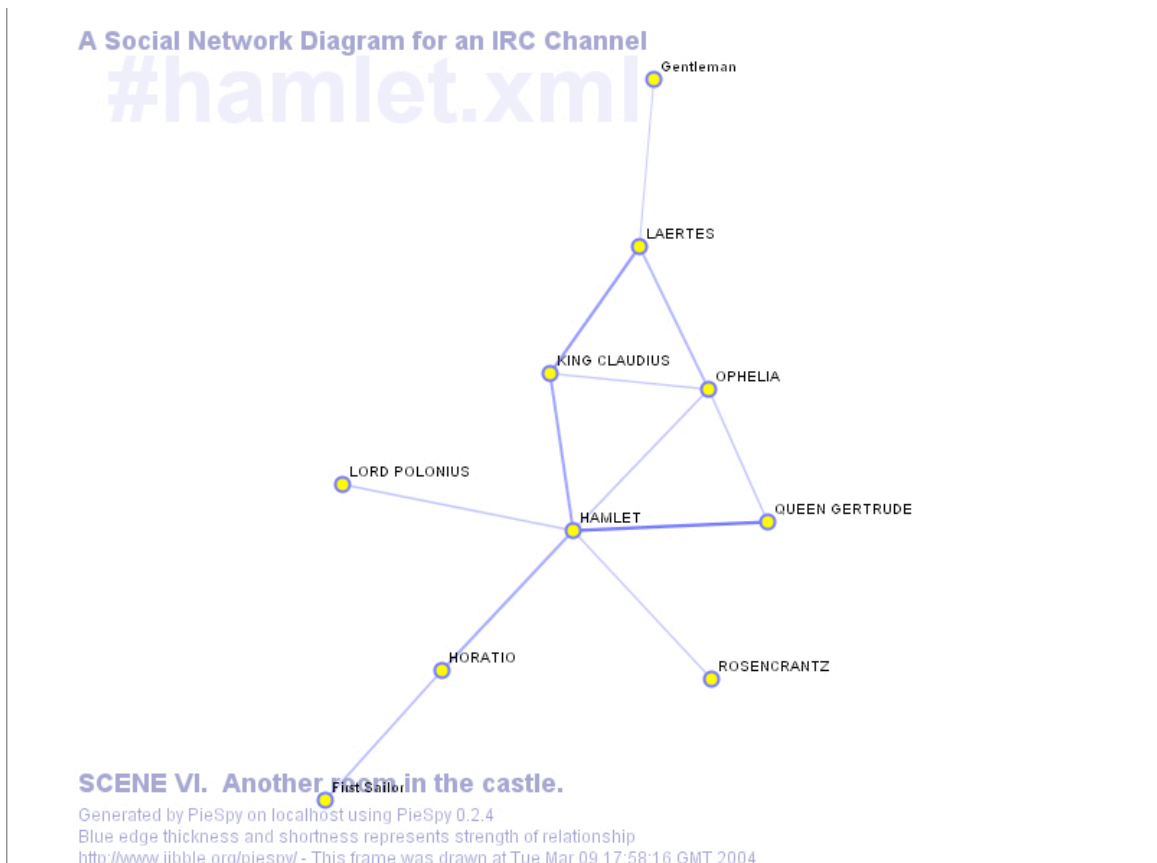


Figure 2.9 – Social Network of Hamlet (Scene VI)

Franzosi’s network analysis of violence in fascist Italy also demonstrates another useful way to use this kind of analysis when combined with a time series. His 2 figures (2010:113) show the ‘star’ network graph for the Sphere of Action of Violence, first for 1919-1920 and then for 1921-22 that clearly show (based on his source data) that the ‘nexus’ of violence shifted from the police at the centre to the fascists at the center for the times series analyzed. In a screenplay context this could be used to show the shifting ‘power’, say, of the protagonists vs. the antagonist as the screenplay progresses, based on their actions or dialog.

Moretti (2011:3-5) also makes a number of useful points about using network models to analyze ‘the hierarchy of centrality’, particularly in relation to the protagonist (he discusses plays but the same principles apply to screenplays). By making a network of a play you create a ‘model’ that you can experiment with. Moretti proposes using this model to test the ‘significance’ or ‘centrality’ of the protagonist by viewing the impact on the model when you remove the protagonist from it or when you remove other characters from it (e.g. the antagonist).

Gibes and Anderson (see <http://clearcongressproject.com/charviz/> - accessed 11 June 2013) have produced a different kind of character visualization that is

‘...accomplished by encoding each typed-character in the screenplay with small bit of visual information - a tiny colored rectangle, the color encoding which character owns that particular portion of text.’

Figure 2.10 shows how this technique can be used to visualize the ‘weight’ of the key characters across episodes of season one of *Buffy the Vampire Slayer*.

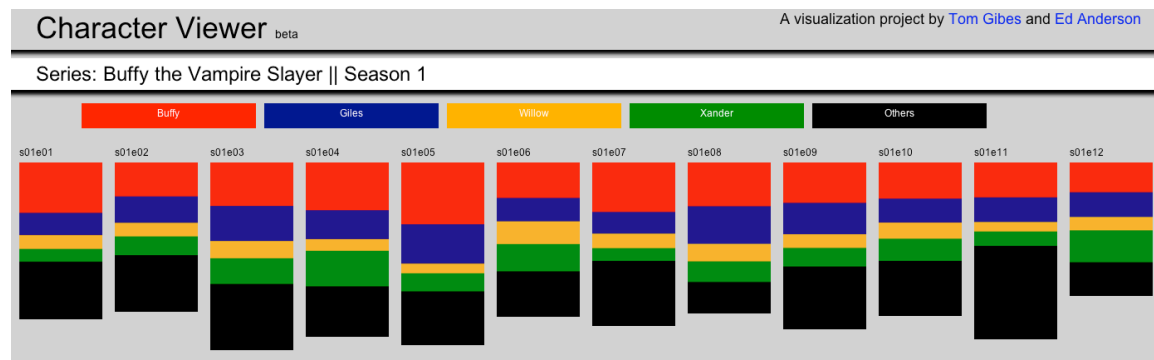


Figure 2.10 – Character Weights in *Buffy*

Conflict and Challenge

We care about characters partly because we are interested in how they cope with conflict and challenge – especially if these characters are in some way “like us” or like we would aspire to be. The more conflict and challenges a

character is involved in, the more likely that character will retain our interest – assuming the conflicts and challenges are realistic and balanced by other kinds of character action. Indick (2004) identifies many kinds of character conflicts exemplified in various movies, including neurotic, normative and existential, and discusses them in psychoanalytic terms in reference to the work of Adler, Erikson, Freud, Jung and May.

Identifying

One way of identifying conflict and challenge in a screenplay might be by linguistic analysis of both action and dialog elements. In the action blocks for example, we might look for the propensity of certain words and terms associated with conflict and challenge – verbs such as “hit”, “shouted”, “pushed”. In the dialog blocks we may look for sequences of dialog focused on two characters to the exclusion of others where the tone is analyzed as confrontational because it contains more statements (e.g. “leave me alone”, “get lost”) than questions.

Visualizing

One way to visualize this aspect of a character in a screenplay is to use a line chart visualization:

- Scenes in which the character is not in conflict or challenged are represented as a straight line or equilibrium.
- Scenes involving the character in conflict or being challenged and responding positively or resolving positively (e.g. high-points”) are represented by more or less “static” above the line.
- Scenes involving the character in conflict or being challenged and responding negatively or resolving positively (e.g. low-points”) are represented by more or less “static” below the line.

What one might look for are patterns of static that also help to visualize the

relative levels of conflict and challenge across the duration of the screenplay.

Choices and Change

The concept of the transformational arc of a character is fundamental to Vogler's Hero's Journey, in which the hero is transformed – usually positively (unless the screenplay is a tragedy) - by a series of experiential stages and encounters. How the character responds to an experience or encounter in terms of the choice(s) made and the change(s) that are manifested in the character as a result of that choice is another aspect of character that may determine whether or not an audience remains engaged with a character throughout the movie,

A choice and any resulting change that occurs may be developmentally ongoing or a one-off event that literally defines much of the way the character behaves going forward. When Michael Corleone chooses his family business over his own wife's wish that he eschews their way of life, he makes a choice that profoundly impacts both his own ongoing development as a character and the ensuing narrative of Francis Ford Coppola's *The Godfather* (1972). Similarly when Josh is temporarily transformed in *Big* (1988, Penny Marshall) from being a child into an adult with a child's worldview, this single event acts as fundamental change that drives the rest of the movie.

Identifying

Murtagh et al. (2008) applied Euclidean Embedding, correspondence analysis and clustering to analyze the feature film script *Casablanca* (1942) and a series of scripts from the US TV show *CSI* (Crime Scene Investigation) to attempt to reveal some aspects of the "deep structure" of the screenplay content. In the case of *Casablanca* this resulted in the identification of pivotal scenes that were indicative of some kind of change in the direction of the narrative. In the case of *CSI*, Murtagh's analysis confirmed that:

'There is occasionally a very strong link between commercial breaks and change in thematic content as evidenced by the hierarchy. In other cases we find continuity of content bridging the gap of the commercial breaks' (2008:17).

Murtagh's approach was guided by story precepts proposed by Robert McKee, who claims that text 'means the sensory surface of a work of art' (1998:252) whereas subtext is 'an inner life that contrasts with or contradicts the text' (1998:255).

Visualizing

If it is possible to identify moments of choice and a subsequent change in behavior then this aspect of character may also be mapped in some way, perhaps through a familiar flowchart visualization using the decision diamond to indicate scenes in which a change point is identified or an 'organization tree' representation of possible choices. This kind of 'branching analysis' could in turn provide guidance to screenwriters as to alternate directions that the narrative may take so these can be more fully understood and explored.

Dynamism

Characters that are merely differentiated by their physical appearance and a collection of traits are likely to be viewed by an audience as "flat" or "static".

Movie audiences don't necessarily want their characters to be wholly realistic, in fact they may prefer them to be larger than life in some way: More decisive, more evil, more lucky. This suggests that dynamic characters who dominate the screen are more likely to have higher audience appeal than more static characters, who appear infrequently and do less. A dynamic character is likely to have a wide social network, to experience relatively more conflict and challenge, to make more choices and exemplify various kinds of change. She is also likely to be more active generally and speak more influential dialog.

Identifying

Being more active implies that the character is associated with more verbs and more active verbs in particular, in scene action descriptions. Speaking more influential dialog implies that he may voice more orders (imperatives) or ask more questions or that more other characters listen to what he says or refer to him/her more often themselves in their own dialog. This kind of analysis requires that the scene action the character is involved in and the dialog he speaks be analyzed in more depth.

Visualizing

One way to visualize character dynamism is to show the relative “footprint” of the character in the screenplay using a bubble graph. A more conventional way is to show a range of characters relative to their ‘position’ against two axes and four quadrants (see figure 2.11).

- Action Axis - more or less active
- Dialog Axis - more or less commanding/questioning/listened to

The more active and commanding/questioning characters are located in the upper right quadrant, which is where one might expect to find the main protagonist/antagonist.

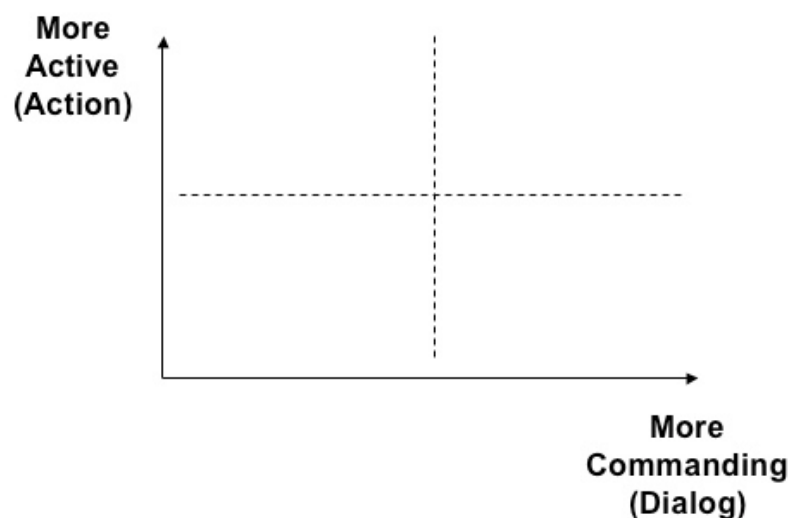


Figure 2.11 - Dynamic Character Matrix

Cohesive Characters

Characters who do not behave consistently in the course of a movie – assuming inconsistent behavior is not a defining trait of the character - may be viewed with suspicion by an audience and eventually cause them to dismiss the character as unbelievable. This could have a direct commercial impact by diminishing the “word-of-mouth” about a movie in a negative way. A consistent character is a cohesive character. Rimmon-Kenan (2002:39) defines the main principles of cohesion as repetition, similarity, contrast and implication.

2.4 Narrative

A story consists of a series of related events, whether that series is ordered forwards or backwards in time or consists of separate sequences of events assembled so as to constitute a viable story. For example, a story about a murder may start with the killing and then look back over the events that led up to it, to “explain it” as a “whydunnit” or progress from the killing event forward to discover the killer and unravel the “whodunnit”. Or like *Rashomon* (1950, Akira Kurosawa) may tell the story from the perspective of different participants involved in the event. In this case the same essential story, that of a murder, is organized by the writer in a different way by using a different narrative approach.

By definition, narrative is about a sequential succession of events. Metz (1974:24) says ‘A narrative is a sum of events; it is these events which are ordered into a sequence... A narrative is not a sequence of closed events but a closed sequence of events.’ According to Rimmon-Kenan (citing Chatman and Barthes 2002:16):

Events can be classified into two main kinds: those that advance the action by opening an alternative (‘kernels’) and those that expand, amplify, maintain or delay the former (‘catalysts’).

She goes on to state that:

Structural descriptions show how events combine to create micro-sequences which in turn combine to form macro-sequences which jointly create the complete story.

In the case of *Rashomon*, it is remembrances from a specific point-of-view of a single event, that create the micro-sequences that in turn create the macro-sequence of the complete story as constructed from the different perspectives.

She cites temporal succession and causality as the two main principles of combination and defines a storyline as a succession of events restricted to one set of individuals. In general screenplay terms, her micro-sequences can be mapped to *scenes* and macro-sequences to *sequences* of scenes; and a specific character's participation in a storyline represents part (or all) of their narrative *throughline*, which itself functions as a macro-sequence.

So can story events be formalized or functionalized? According to Propp, in the world of the Russian folktale, they can. Metz considers Propp to be a structuralist interested in structural analysis and comments (1974:16) that 'it might be said that the main interest of structural analysis is only in being able to find what was already there, of accounting with much more precision for what a naïve consciousness had "picked up" without analysis.' Propp's *Morphology of the Folktale* calls the constant elements of a folktale "functions" (1968:21) and states that, 'Function is understood as an act of a character, defined from the point of view of its significance for the course of the action.' Propp proposed that these functions could be combined into 31 folktale patterns. But this rather mechanistic analysis was later challenged by Bremond's (1966:75) existentialist, process-centric perspective based on the idea that three functions combine to form a sequence of logical steps: potentiality, process and outcome. This potential could result in either an actualization or non-actualization process and the outcome of the actualization process result in success (improvement) or failure (deterioration).

Greimas (*Structural Semantics* (1966) referenced in Edgar 2009:22) also applied reductive techniques to characters. He defined six plot functions or 'actants': Sender/receiver, subject/object and helper/opponent. Actants truly represent 'roles' that carry out a function in the plot rather than individual characters i.e. the role has an arc rather than the character representing the role. On this basis, Greimas found three plot forms in folktales: Contractual, performative and disjunctive. All this indicates that structural analysis of creative works has a long pre-Fieldian history.

The fundamentals of screenplay narrative analysis assume the following can be easily identified from a screenplay.

- The extent (start and end) of the screenplay.
- What plots/sub-plots/storylines interweave to create the narrative line.
- When these storylines start and end (their extent).
- What sequences of scenes are represented in the narrative as a whole, a plot or sub-plot.
- Where plot intersects with scenes that are part of the throughline of specific characters in the screenplay.

Plots, storylines and sequences are not usually "flagged" in the text of the screenplay; in fact in initial drafts a screenwriter may not even be aware they exist – only identifying them and rejecting or refining them in the revision/rewrite process when some kind of structure is applied to the text manually or surfaced from the data automatically. Therefore identifying plots and/or sequences within a script requires some work by the human analyst or software analytical engine, first to identify them and then to visualize them.

Linear and Non-Linear

Most screenplays have a linear narrative, although non-linear narrative became fashionable following the critical and box-office success of Tarantino's *Pulp Fiction* (1994). Linear narratives follow a sequential, time-based start-to-end flow (or end-to-start flow) whereas a non-linear narrative jumbles up the narrative so that the narrative may jump from past, to present, to future in a non-sequential way (e.g. *The Machinist* (2004, Brad Anderson)). However, a linear narrative may also include either flashback and flash-forward scenes or sequences that disturb but do not essentially alter the linearity of the narrative. For example *Titanic* (1997) that comprises largely one big flashback, bookended by scenes located in the present. But a non-linear narrative does not have to start at the beginning of the narrative in question e.g. *Memento*, (2000, Christopher Nolan) and may backtrack or jump ahead in ways that disrupt the sequential flow of a traditional linear narrative.

Non-linear narrative form is becoming more important as the 'gamification' of stories, to create online interactive narratives that leverage the full potential of in-story hyperlinking, and interactive documentaries or 'i-docs' become more popular. Tools that support the creation of non-linear narratives like Florian Thalhoffer's *Korsakow* system (korsakow.org), Mozilla *Popcorn Maker* (popcorn.webmaker.org) and *Storyplanet* (storyplanet.com) for example, enable creatives to combine text, links, images and audio/video clips in ways that call into question the traditional concept of a screenplay or even the need for one at all. In effect the screenplay is not used to drive the production of the interactive narrative but needs to be extrapolated from it, after-the-fact.

However, visualizing the narrative line in linear narratives is useful because it may:

- indicate gaps or breaks in the story
- expose narrative overlap or lack of it

- highlight scenes that need to be moved, removed or added

Any of which may cause an audience to become confused or irritated by the resulting movie, reducing the potential for positive word-of-mouth recommendations.

Identifying

The narrative line is identified by the order in which the scenes in the screenplay are presented. Individual scenes may be part of a single narrative line or multiple narrative lines. But in the latter case, it is not necessarily easy to identify where one narrative line ends and another begins (or continues). Although this may be more evident because each narrative line takes place in the same milieu or within a specific timeslot or involves a specific set of characters who do not cross narrative lines (e.g. in a portmanteau movie).

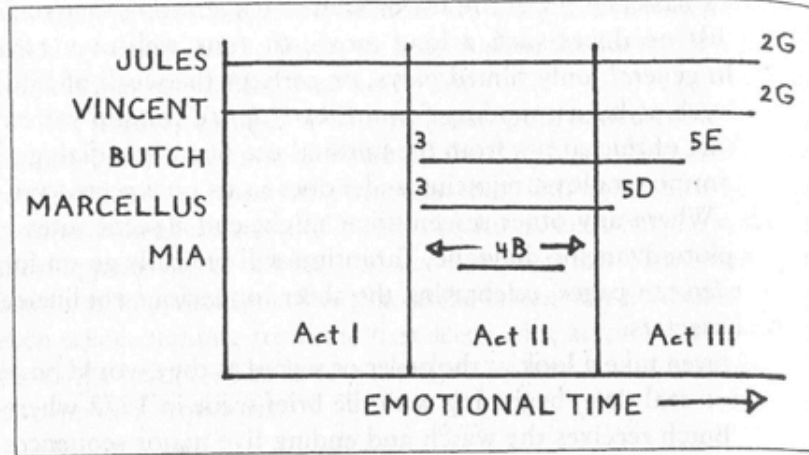
Visualizing

Visualizing a linear narrative is relatively straightforward to analyze and visualize. Often the analysis merely needs to identify scenes linked to a role, sequence or storyline and visualization takes the form of scenes listed sequentially along a timeline that references the anticipated running time of the movie based on the page count of the screenplay. If there are multiple narrative lines then these can be visualized using “swim lanes” whereby each narrative line and the scenes that represent it are contained within their own lane.

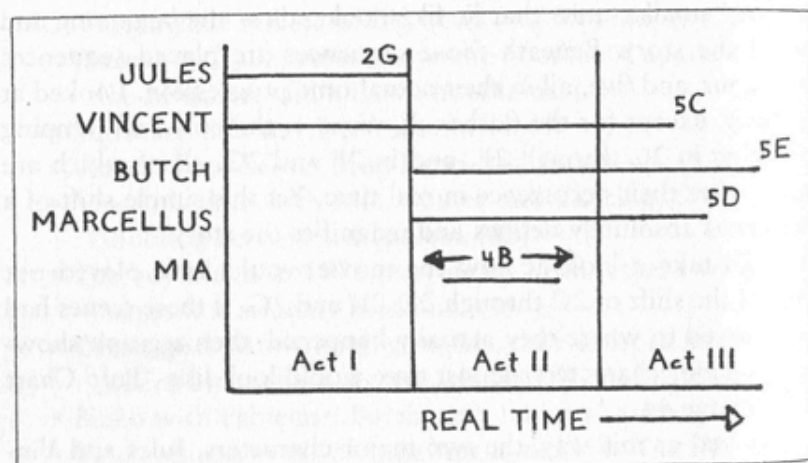
But any visualization becomes more complex when the screenplay involves multiple narrative lines organized non-linearly or makes extensive use of flashback/flash-forward over the course of the screenplay. In these cases there may be multiple narrative “strands”, which may overlap each other and narrative “jumps” both backwards and forwards.

Figure 2.12 shows Pope’s (1998:44) analysis of *Pulp Fiction* with multiple

character throughlines being reorganized by director Tarantino to emphasize rising emotional tension rather than the linear way that events actually unfolded.



Pulp Chart #1



Pulp Chart #2

Figure 2.12 - Pulp Fiction Narrative Lines

Figure 2.13 shows Aronson's (2001:113) analysis of the narrative structure of Shine involving the interlocking connections between the "film in the present" and the "film in the past".

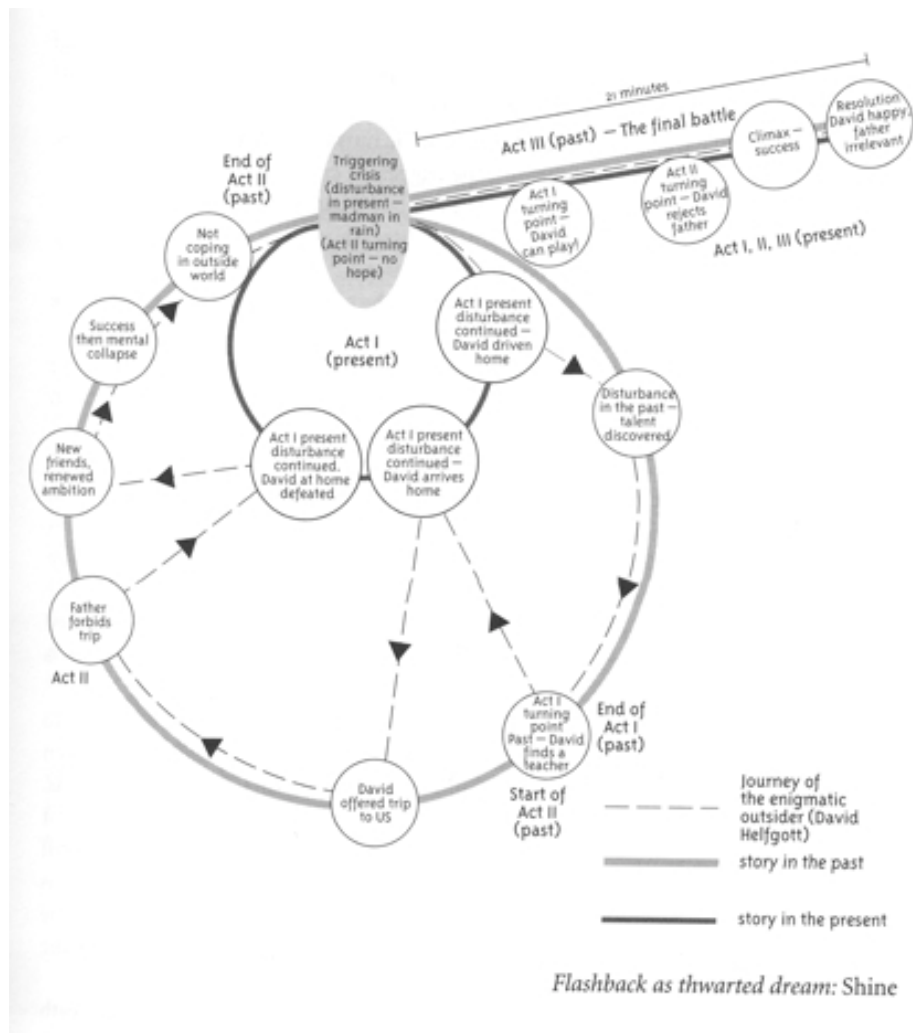


Figure 2.13 - Flashback as Thwarted Dream: Shine

Plot Points

Many screenwriting “gurus” advocate the use of defined plot points such as “turning points”, “inciting incidents” “midpoints” and “climaxes” to punctuate and guide the screenplay narrative. These plot points often signal a significant action or change in direction within the narrative. Apparently some script readers actually look for these plot points and expect them to occur on or around specific pages in a script. So a visualization of these plot points would essentially present this information automatically before the script is read, saving valuable script-reading time to find them manually (except of course if readers are so focused on this approach that they simply skip to certain pages in a script to find them).

Identifying

Currently screenwriting programs generally do not provide a way to identify plot points in the script. So again, identifying the plot points may require the writer to specifically tag them in the content either as they are written or after the fact, in the absence of screenplay parsing software sophisticated enough to find these plot points and tag them automatically as one kind of screenplay metadata.

Visualizing

Plot points can be visualized as a series of peaks and troughs, rollercoaster fashion. One might expect at least 3 such peaks in the average 3-act movie namely: The inciting incident (in act one), the mid-point (in act two) and the climax (in act three). The kind of structure reflected in Freytag's Pyramid (see figure 2.1) where (a) is the *introduction*, (b) is the *rise*, (c) is the *climax*, (d) is the *return or fall* and (e) is the *catastrophe*. The difficulty is identifying them, without the writer signposting them by explicitly tagging their occurrence in the script.

2.5 Structure

Acts, Sequences, Scenes and Beats

All screenwriters' practice scene writing and many will break a scene down into "beats" and/or combine scenes into "sequences" and "acts". These may be regarded as the core structural components of the screenplay narrative and represent a hierarchy of increasing detail such as:

3-5 acts comprising

10-20 sequences comprising

50-100 scenes comprising

150-300 beats

Some writers may go as far as to write scenes where each scene itself has an identifiable beginning, middle and end and could also employ this triad structure again when composing sequences of scenes to represent a logical dramatic unit within the screenplay. In this sense the whole screenplay becomes an example of a fractal design.

Identifying

Whilst scenes are always identified in a screenplay – by means of the scene heading or slug line - acts, sequences and beats are not. Unless, that is, the screenwriting package used by the writer provides the ability to organize scenes and the content of a scene in this way. So identifying the scope of these structural components may require the writer to specifically tag them in the content either as they are written or after the fact.

Visualizing

This kind of visualization is better suited to navigation of screenplay content rather than analysis. It is useful to see the “forest and the trees” to navigate and jump-around the script content and to use as a means to “shuffle” and reorganize the script content, say by dragging a scene from one sequence and dropping it into another. It is also useful for understanding if the script is “well-balanced” and does not deviate too far from accepted norms – a script with 20 acts and 500 scenes would be perceived as unbalanced for example.

Balanced Acts

Proponents of the three-act structure, including script readers who are anecdotally supposed to look for the three-act structure as a basis for commercial movie scripts, are quite exact about the “balance” between the content of the acts. Field (see figure 2.12 below) divides a screenplay into roughly one quarter (act 1), two quarters (act 2) and one quarter (act 3). In this context an unbalanced act structure may make a screenplay less commercially acceptable.

Also relevant here is Yanno's concept (2006: 15) of act 1 asking a question and act 3 answering it - an act-based setup and payoff. If act 1 asks a question that act 3 does not answer or if act 3 answers a question that act 1 does not ask, the screenplay may also be unbalanced.

Identifying

Balance depends on the ability to detect act breaks that are unlikely to be specifically defined by the screenwriter in the script. The question/answer balance depends on identification of attribute in the part of the script identified as act 1 that can be found to have been referenced again somehow in the part of the script identified as act 3.

Visualization

A linear representation of script balance seems the best solution to show balance between the three acts but visualizing the question and answer/setup and payoff balance between act 1 and act 3 is more difficult. One simple option is a two column display headed with potential questions listed in the left column (act 1) and potential answers listed in the right column (act 3). Questions with no answers or answers with no questions would then be easy to identify, allowing the writer to revisit the appropriate act and rewrite to put the screenplay back in balance.

Scenes – Similarity and Difference

The scene is the key structural entity within a screenplay because it can both be decomposed into shots and aggregated into sequences or storylines. An interesting approach to scene analysis is to look for similarities and differences between a set of scenes, or all scenes in the screenplay, and what this might tell us. Scene analysis can refer to either data or metadata or both. One obvious similarity/difference is whether a scene is internal (INT) or external (EXT) and whether it is DAY or NIGHT - others include:

- Do the scenes take place in the same location?

- Do the same set of characters take part in the scenes?
- Does one specific character appear in more/less scenes than other characters?
- Are the scenes action only or action and dialog?
- Is the linguistic “footprint” of the scenes similar?

Consider Ridley Scott’s *Alien* (1979) in the light of the above.

- Most scenes take place in the spaceship except around the beginning (on the beacon-sending planet) and at the end (in the escape shuttle). These differences may be considered important in leading to and from the core “middle” of the movie. The propensity of scenes in the same place is also something that is likely to be characteristic of a specific genre e.g. horror or a sitcom.
- Many scenes that involve Ripley (Sigourney Weaver) peripherally involve other crewmembers being killed by the Alien; whereas Ripley is present in the majority of scenes. From this one might deduce that she is probably the protagonist, as she not only has a throughline from the beginning of the movie until the end, she also has a “lifeline”.
- Many scenes that involve Ripley and one other character – even if that character is “mother” the ship’s computer – are usually scenes where Ripley challenges the other character in some way. Again, indicative of protagonist (or antagonist) behavior.
- In the last third of the movie, Ripley appears in many “action-only” scenes – unsurprising, as the rest of the crew is dead. But this is a big difference from earlier scenes that are relatively dialog-heavy as the whole crew discuss their situation and argue about how to kill the monster. By identifying the change in scene type, we can locate some kind of “tipping point” that probably occurs around the time that the

last crewmember is killed by the Alien.

- The linguistic footprint of the scenes up until the last crewmember dies may possibly indicate rising tension or fragmentation of the team under pressure. Perhaps this could be reflected in more statements than questions, more expletives, more action than dialog.

Wherever the Alien appears in a scene, death often occurs also. This could be indicative of an antagonist role, especially as towards the end Ripley (possible protagonist) and the Alien (possible antagonist) are essentially the only characters left in the screenplay (apart from the cat that is).

2.6 Analytic Archetypes

Significant academic attention has been focused on using “intangible” screenplay text as an input to automatically generate some kind of “tangible” animated output or as the raw material to auto-manufacture a computer-generated visual product.

Zhi-Qiang Liu’s ScriptViz takes well-formed sentences from a script as input and converts these into agents and plans for those agents that are then rendered by a “scene generator” in the form of computer-generated animations as the story evolves. Coyne and Sproat’s WordsEye system converts text input into scenes by tagging and parsing sentences to convert them into a dependency structure that is converted into a semantic representation. This representation is subject to depiction rules that generate a series of depictees to which transduction rules are applied to create the refined depictees that are used to manipulate the 3D objects used to finally render the scene to the user as a computer generated animation.

Ramirez proposed using a film language (FL) based on Arijon’s *Grammar of the Film Language* (1976) to generate an ontology from a given script input that could then be interrogated by a reasoning engine as the basis for analysis and generation of a computer-generated cinematic version of the story. Whereas Johsi, Wang and Li’s Story Picturing Engine is more of an

automated storyboarding application that first identifies keywords in the text, then searches an annotated image database for suitable images to match with the text and then determines which images are the “elite” images to illustrate a story using a mutual reinforcement ranking.

The rise of computer gaming as an interactive rather than traditional narrative has focused attention on interactive storytelling and ways of generating visually interactive products from texts or using existing stories and narrative paths to generate new stories or alternative narrative paths within a story.

Gervás, Díaz-Agudo, Peinado and Hervás utilized case-based reasoning based on an ontology that leverages Propp’s *Morphology of the Folktale*, to generate a new plot sketch based on the plot structures extracted from existing stories submitted by the user.

However, surprisingly few screenwriting manuals either advocate or use analytics and visualization as a means to improve a screenplay. But, as suggested above, even the limited visualizations of screenplay content that are presented are almost certainly not generated from analyzing the script itself (the input) but reverse-engineered from watching the movie that resulted from the script (the output).

A review of screenwriting literature indicates that there is only a handful of content visualization archetypes represented, defined here as: Linear, Circular, Nodal and Rollercoaster. Samples of each archetype are displayed and discussed below.

Linear Archetype

The linear archetype can help to visualize the narrative structure and the chronology of the screenplay. Linear visualizations are useful for viewing timelines and throughlines or the narrative threads that relate to characters or plots/subplots in the screenplay.

The simple example in figure 2.14 below visualizes Field's (2005:21) 3-act "paradigm" against the timeline of a 120-minute feature-film. Field maps Aristotle's Beginning, Middle and End structure to his own Set-Up, Confrontation and Resolution. He even provides suggested page extents for each of the three acts.

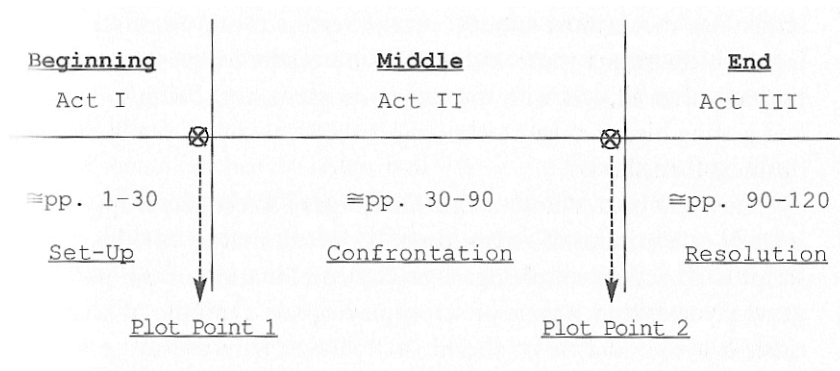


Figure 2.14 - Field's Three-Act Structure

Figure 2.15 below indicates how the linear archetype can also be used to visualize parallel, converging or intersecting "throughlines" as in the screenplay of *The Day of the Jackal* (1973, Fred Zinnemann) that help to generate the tension that reflects both the 'closing of the net' around the Jackal and the Jackal getting ever closer to his assassination target.

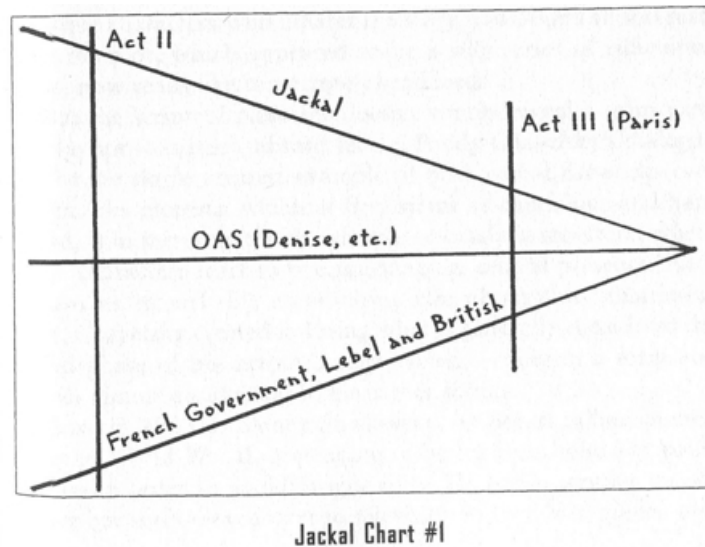


Figure 2.15 - Converging Plot Lines in *The Day of the Jackal* (Pope, 1998:173)

Linear archetypes, that structure screenplays into acts and sequences are not well represented in today's screenwriting software, perhaps indicating that this archetype is not considered to have a practical use for writers.

Circular Archetype

The circular archetype is less suitable for chronological timelines and more suitable for visualizing emotional or psychological "journeys" with a holistic dimension that includes a return to the start, implying psychological resolution or re-integration. Homer's *Odyssey* is the story archetype. The character whose journey the circle reflects may in fact remain in or around the same physical place throughout this kind of journey. Each stage of the journey could "map" to a particular sequence of scenes, or a key individual scene in the screenplay. The circular archetype is more suitable for visualizing character development and is exemplified by Vogler's Hero's Journey (from his 1992 *The Writer's Journey* see figure 2.16), which may have been borrowed from Murdock's 1990 *The Heroine's Journey* (see figure 2.17).



Figure 2.16 - Vogler's Hero's Journey (Vogler, 1994:194)

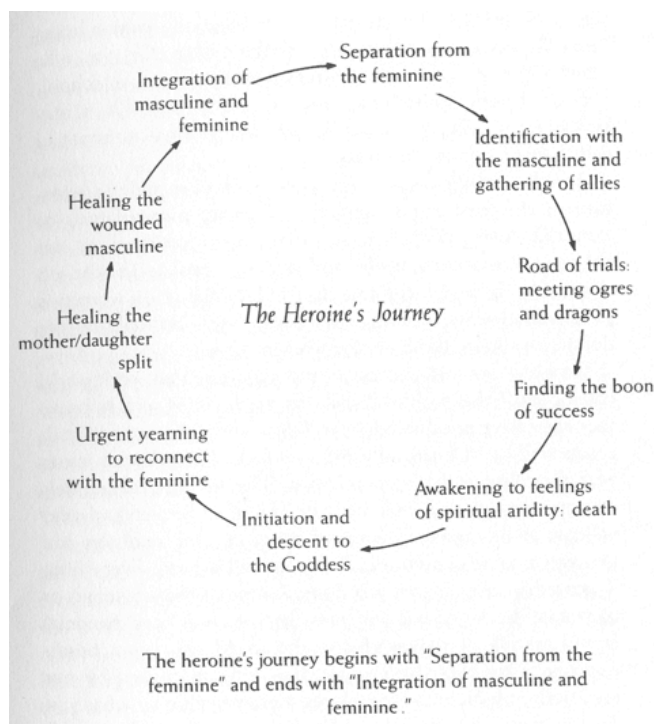


Figure 2.17 - Murdock's Heroine's Journey (Murdock, 1990:5)

Some useful examples of using the hero's journey version of the circular archetype for movie analysis are found in Voytilla (1999). Figure 2.16 shows his analysis of the movie *Die Hard* (1998), which maps plot points onto the

timeline of the movie to show how McClane moves from his ordinary world to the special world of the terrorists and back to his ordinary world.

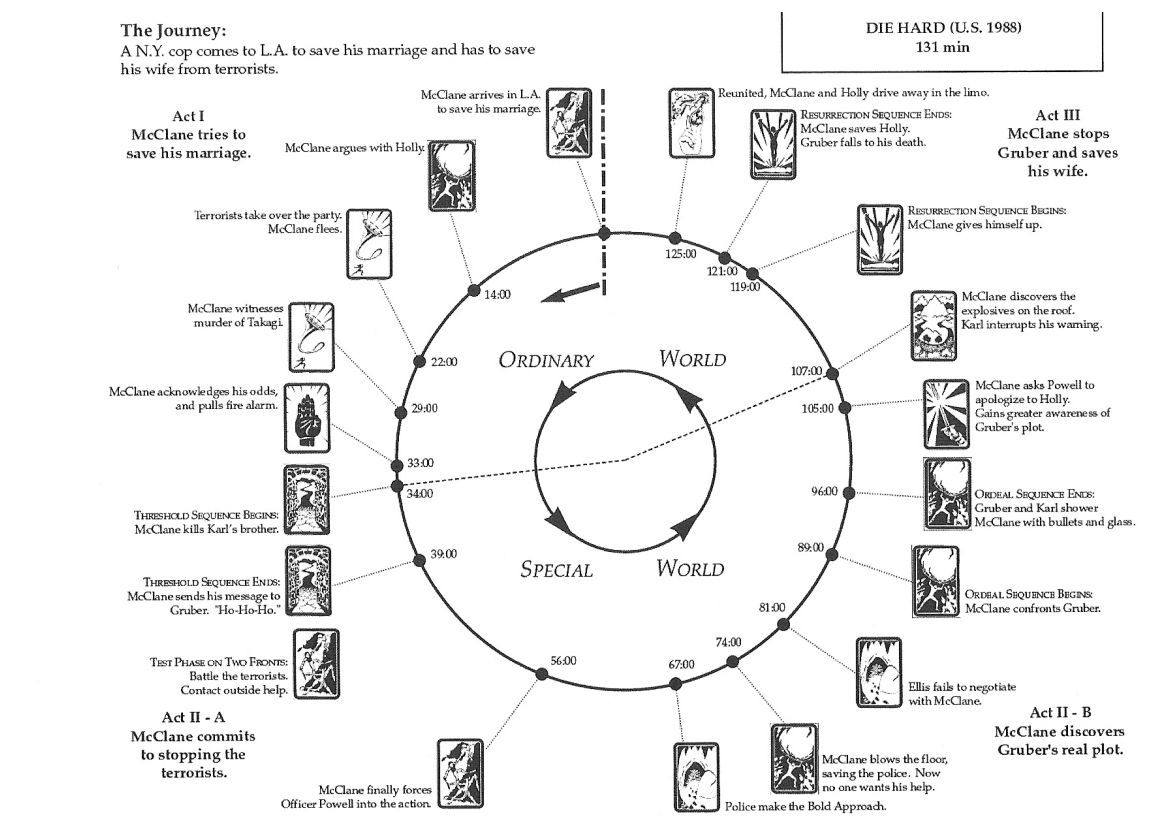


Figure 2.18 - Die Hard Analysis (Voytilla, 1999:41)

Rollercoaster Archetype

The “rollercoaster” archetype (see figure 2.19a below) reflects that fact that the narrative of a screenplay often delivers a series of peaks (or emotional climaxes) and troughs (or releases from tension) for the movie audience; involves creating escalating tension that demands resolution in a satisfying climax; and as the movie progresses, more story information is provided and the characters developed so that the “bulk” of the movie “mountain” increases until it reaches its “summit” at the climax of the movie. Cooper (1997) uses the rollercoaster term whereas Aronson (2001:42) prefers the term “mountain” and Karetnikova (1990:32) simply states, “It is always helpful to view the development of a plot graphically as a kind of triangle, in which the angles represent the opening, the climax and the resolution.”

Throughout her book, Cooper discusses a variety of rollercoaster examples that offer an interesting way to visualize plot points in particular, with both the “peaks” and “troughs” providing a useful way to indicate “highs” and “lows” in the flow of the script and the bulk of the script text providing the means of showing the “information mountain” contained in the script building over time in the mind of the audience. She also discusses the use of rollercoasters as a tool for writers to proactively design a specific kind of narrative experience for the audience.

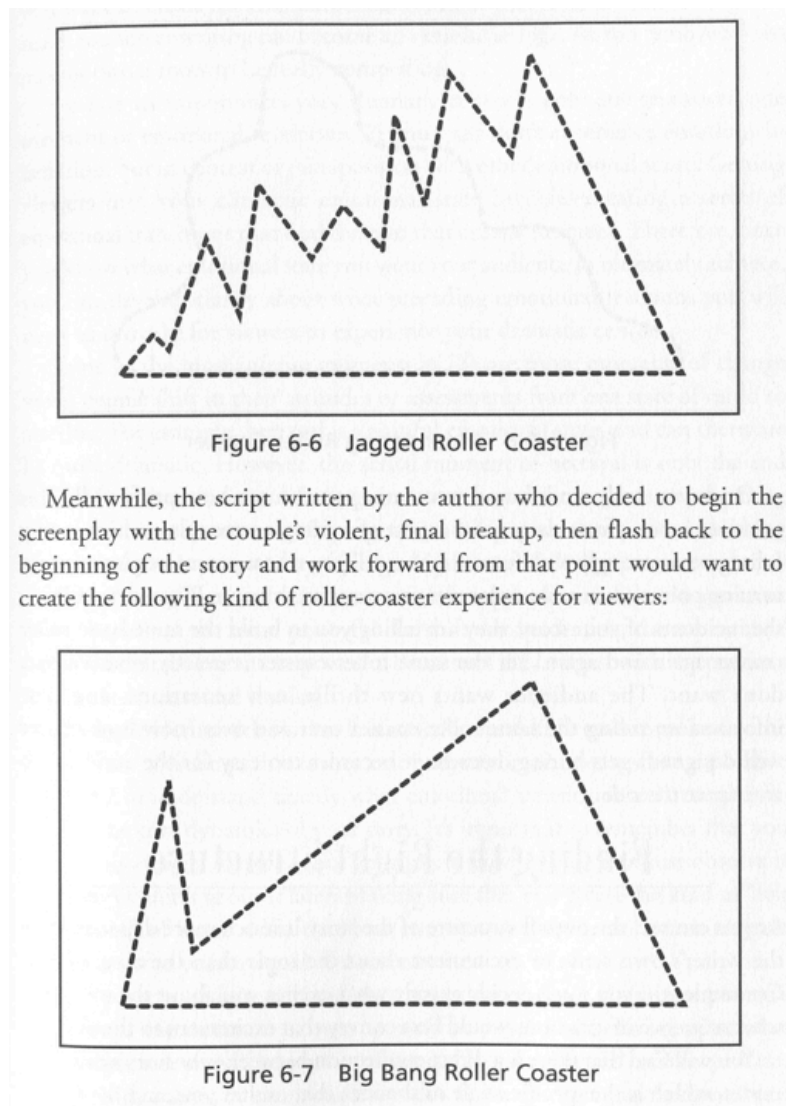


Figure 2.19a - Rollercoasters of Escalating Tension (Cooper, 1997:71)

However, as Aronson herself points out (2010: 293) the mountain visualization can break down when applied to characters within movies

with an unconventional structure such as Paul and Jack’s stories in *21 Grams* (2003, Alejandro González Iñárritu), resulting in a structure that she terms a ‘fractured tandem’.

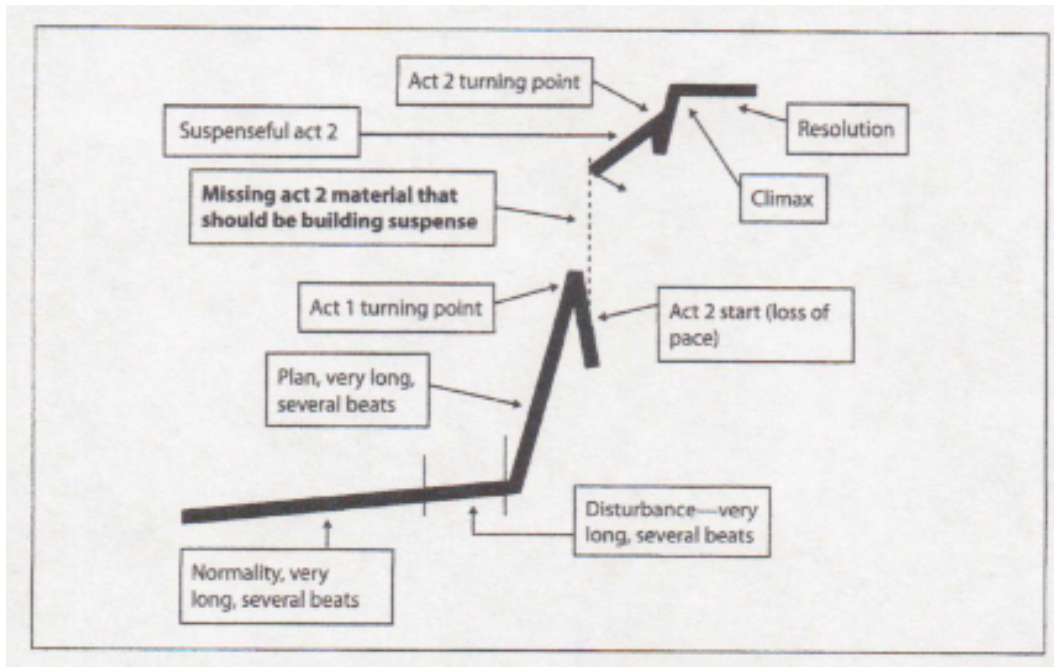


Figure 2.19b - Paul and Jack’s Fractured Tandem in *21 Grams* (Aronson, 2010:293)

Nodal Archetype

The nodal archetype offers a number of implementations. A common form is a top-down or left-to-right, tree-and-branch visualization to show hierarchies or parent-child dependencies (vertical or horizontal dendrograms). For example, figure 2.20 shows an “organization chart” view of the tree structure of a screenplay to visualize the relationship within a single script between acts, sequences, and scenes. This archetype is also useful as a way to visualize parallel or multiple storylines, especially where these storylines are nested within each other. Each storyline becomes a branch with its own set of dependent nodes that may or may not be nested within another branch. Equally the branches could be literally that in a dendrogram of an interactive narrative where each branch represents a different path through the narrative that the viewer/player can take.

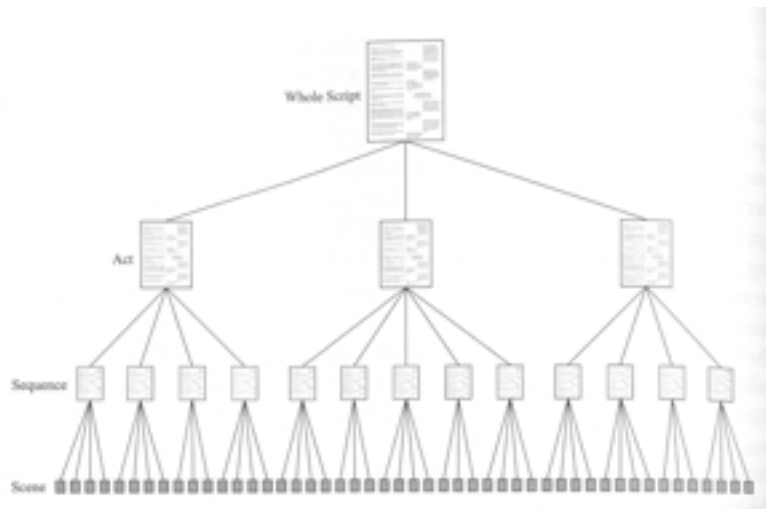


Figure 2.20 - Screenplay Structure as Node Tree (Kitchen, 2006:162)

Another nodal implementation relates to character analysis, whereby each character acts as a node in the relationship network of the script's characters. Figure 5.21 shows a visualization of the character relationships or social network of *The Third Man* (1949).

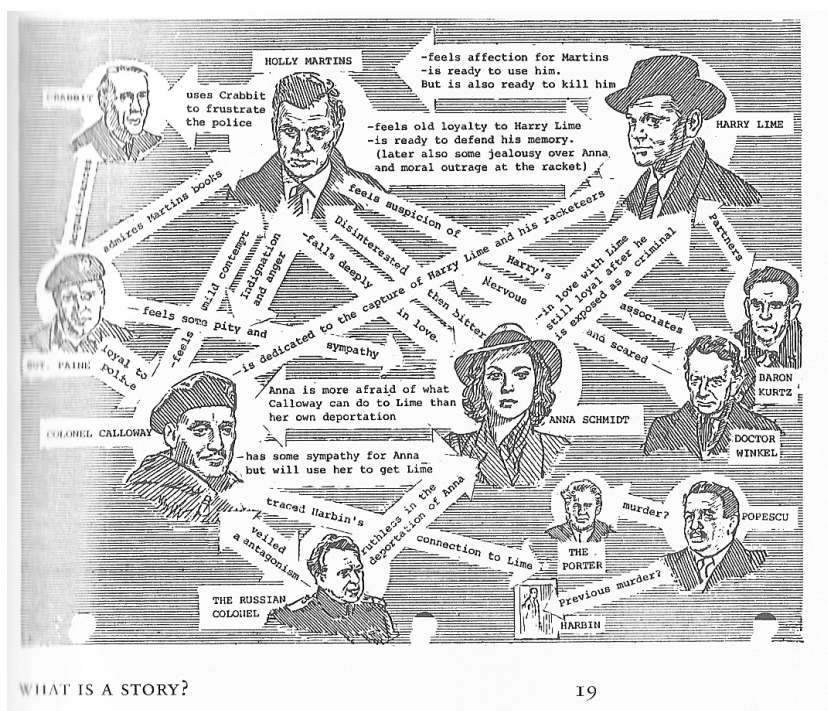


Figure 2.21 - Characters in *The Third Man* (MacKendrick, 2005:19)

2.7 Generic Data Analysis and Visualization Tools

The screenplay visualization examples above, drawn largely from screenwriting 'how-to' manuals, were almost certainly all created manually by the authors - as opposed to by clicking a button in a screenwriting application. But there are generic analysis and visualization tools available that can generate interesting visualizations when provided with screenplay content as a data input.

There are dozens of chart formats that could be used to display screenplay as data content visually (see <http://mbostock.github.io/protovis/ex/> - accessed 11 June 2013) for over 50 chart types that can be produced from the Protovis tool. This section discusses some software tools that have been designed for generic data analysis and/or visualization that interesting to apply specifically to screenplay content. The first three of these visualizations utilize tools are found on Jeff Clark's web site at www.neoformix.com. The example visualizations are produced from my own script *Carpini*.

Arc Diagram

According to Clark:

Document Arc Diagrams illustrate the similarity structure within a text document by drawing arcs connecting segments of a document that share similar vocabulary...I have written before about Martin Wattenberg's Arc Diagrams for visualizing structure within strings. They are an intriguing way of visualizing repetition at varying scales within a linear sequence.

Clark explains the process of generating an arc diagram as follows:

1. Break the document up into a stream of words.
2. Throw away any 'stop words' (a, at, of, the...).

3. Divide the remaining stream of more interesting words into 50 equal segments based on linear position.
4. Calculate a similarity metric between each pair of segments based on the amount of overlapping words.
5. Draw a diagram where the document segments are connected by arcs with the transparency determined by the similarity between the segments. Use a threshold so that weakly connected arcs don't get drawn at all.
6. Show the top two words for each arc drawn at both segment endpoints.

This visualization could be more interesting when applied to a screenplay, assuming the following modifications:

1. If the squares (see step 3 above) represented each scene in the script, rather than an arbitrary 50 segments, the diagram would have a more screenplay friendly structure.
2. If the character names included in each scene and/or scene headings were listed against each square consistently, this would be more useful.
3. If the text included could be selected as just dialog elements, just action elements or both, rather than all text it might be more interesting.

If the language content could be defined in some way e.g. conflict language, the arcs could be colored to reflect this and used to connect scenes of similar linguistic type. I used one of my own scripts, *Carpini*, to convert into a suitable data input in order to generate the three diagrams shown in figure 2.22-2.24 below. Combinations of character names are listed to the left of the 'scene square' and the curves represent links between scenes deemed to be 'similar' (the orange line indicates a stronger similarity than the blue

line).

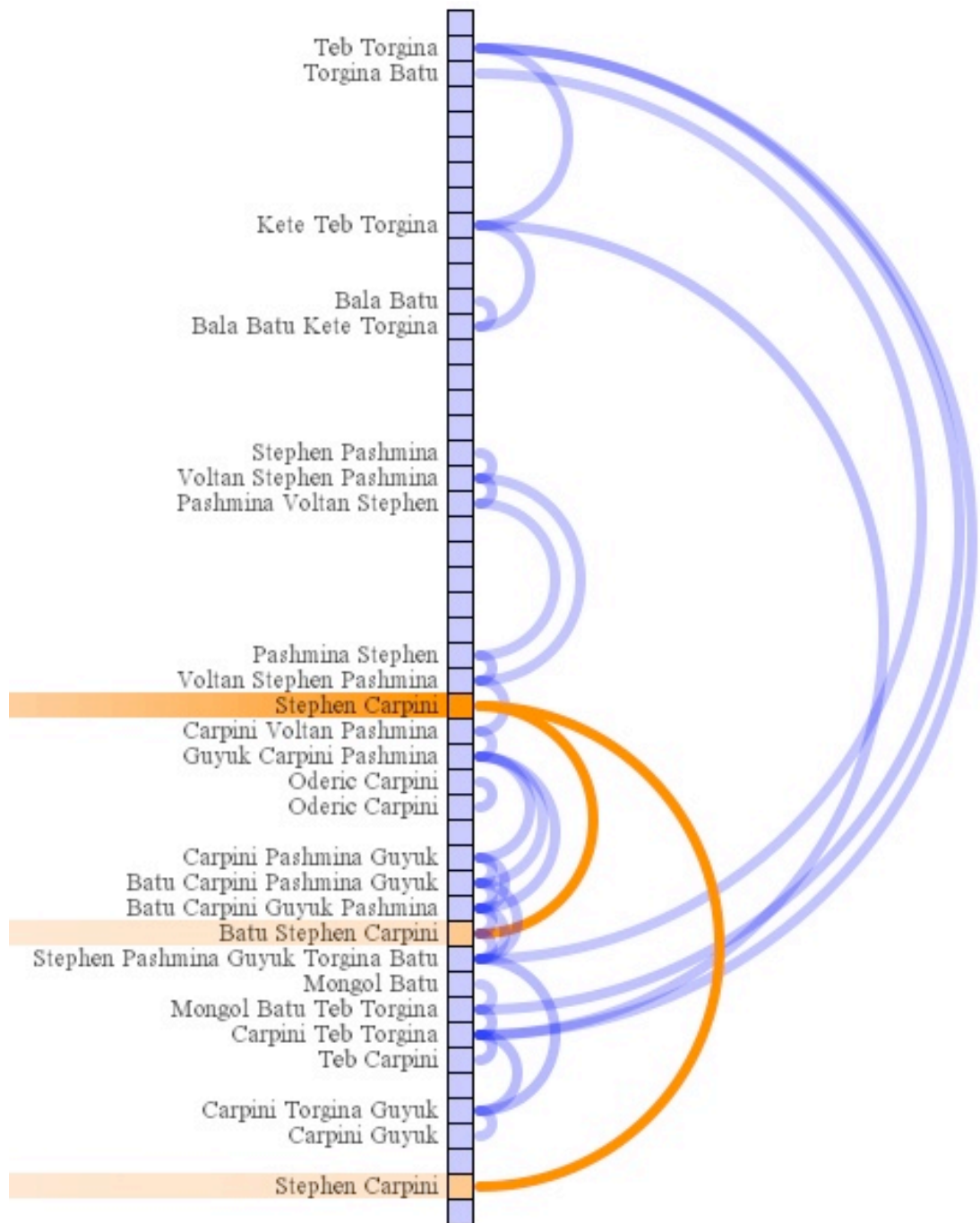


Figure 2.22 - Arc Diagram Applied to my Screenplay Carpini

Topic Flower

The topic flower presents an interesting and artistic tonal impression of a body of text that reflects the occurrence of certain defined categories of language in the text. Here the topic flower is designed to reflect the content

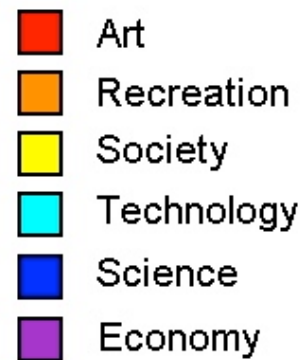
in terms of the relative weight of language relating to Art, Recreation, Society, Technology, Science and Economy – categories clearly more relevant to news or factual content rather than a screenplay.

Clark explains the process of generating a topic flower visualization as follows:

1. The same text will always generate the same flower.
2. More text will generate more layers of petals.
3. The primary topic will be shown using the associated colour on the outermost two layers of petals.
4. If there is a secondary topic it will be shown on the third layer of petals. This pattern repeats, two layers using the primary, then one with the secondary.
5. If there exists a tertiary topic its' colour is used to accent the edges of some of the primary coloured petals.
6. The number of little 'hairs' on the flower is indicative of the number of personal pronouns used in the text.
7. Rounder petal shapes are suggestive of emotionally positive terms (love, yes, peace), and more elongated terms indicate negative terms (death, murder, idiot).

This visualization could be interesting if applied to a screenplay, assuming the following modifications:

1. If the topics were changed to reflect more screenplay relevant factors such as emotion or sex or violence or conflict – the linguistic categories would be more relevant.



Source code: [sketch_flower0](#)

Built with [Processing](#)

Figure 2.23- A Topic Flower Applied to a Screenplay

Double Document Shared Word Diagrams

According to Clark:

Double Document Shared Word Diagrams compare and contrast two documents by showing both the unique and shared vocabulary and it's distribution across the two documents of interest.

The two columns of squares represent the two documents. The leftmost column of word circles shows the highest frequency non-trivial words found in document 1 but not document 2. The rightmost column of word circles shows those words unique to 2 and the central column shows the words that are common to both.

This visualization could be interesting if applied to a screenplay, assuming the following modifications:

1. If the squares represented scenes in the script, the diagram would have a more screenplay friendly structure.
2. If character names common to both drafts were listed exclusively in the centre column, this would be more useful.

3. If screenplay specific noise words such as “INT” and “EXT” were filtered out automatically.

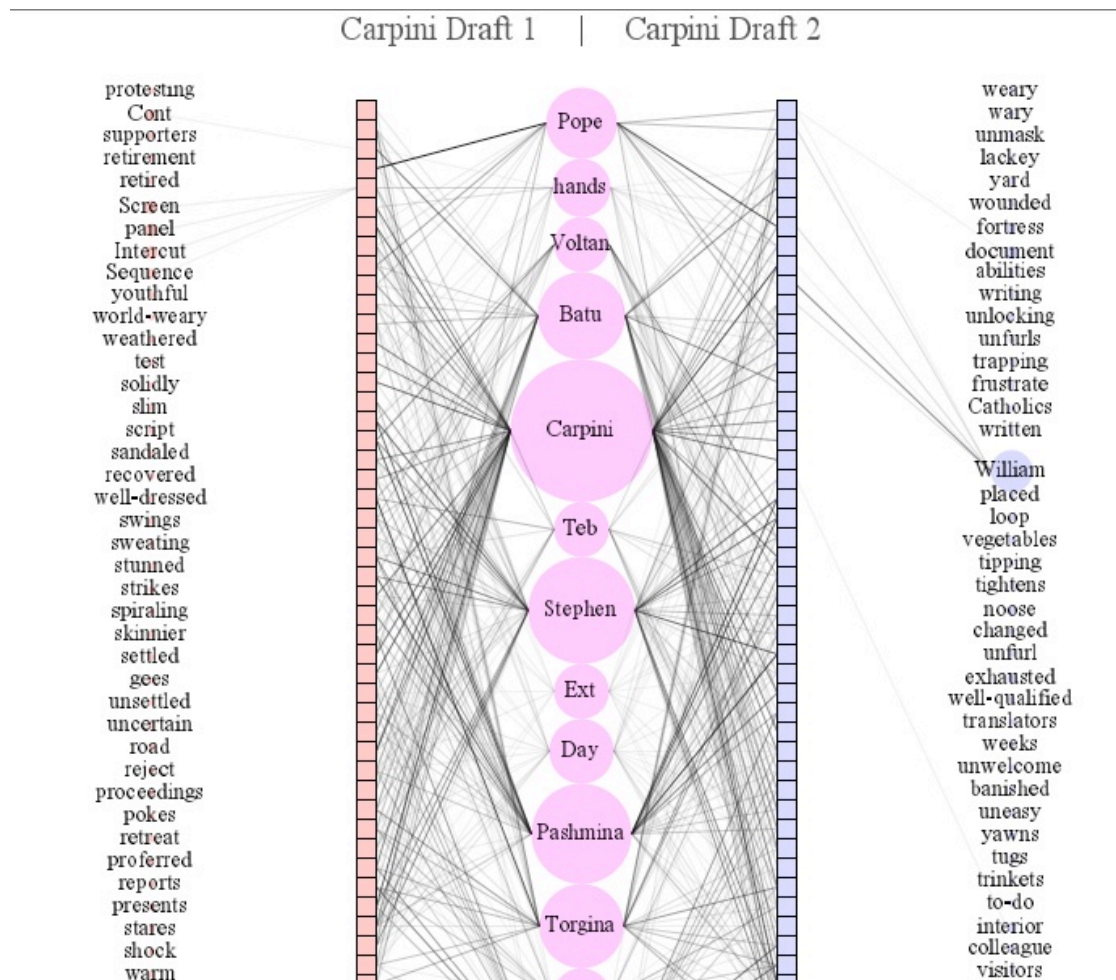


Figure 2.24 - A Shared Word Diagram Applied to Two Drafts of a Screenplay

Dialog Analysis

Of course it is now very easy to search any digital screenplay for specific words or phrases to locate potential product placement opportunities or to identify profanity and sexual terms in dialog, which could impact the potential movie rating. But dialog analysis is also interesting in the context of the interplay between specific characters in a screenplay or in relation to what is realistic for the time-period the script is set in.

Some interesting visualizations have been produced to analyze the U.S. presidential primaries in the run up to the 2008 election. In screenplay terms, most of these have been dialog-centric, that is they have focused on the spoken content of the candidate debates and speeches.

(<http://www.nytimes.com/interactive/2007/12/15/us/politics/DEBATE.html> accessed on 06 July 2008)

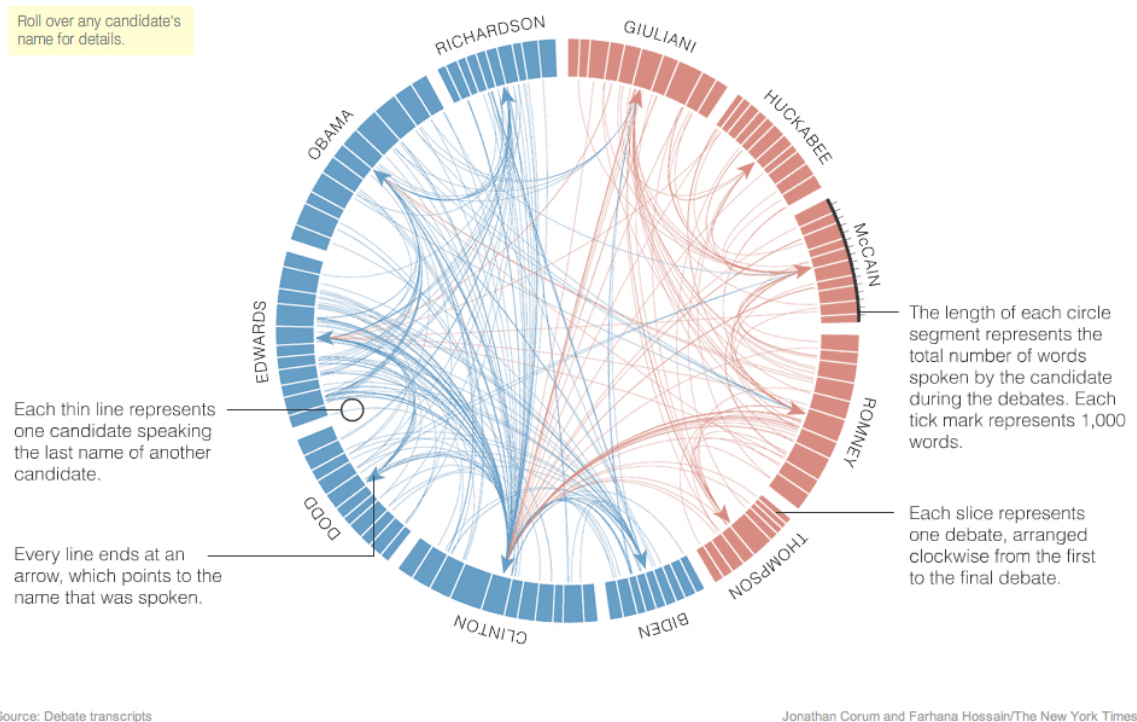


Figure 2.25 - Naming Names

Figure 2.25 was produced by *The New York Times* (15 December 2007) and shows the names referenced by major presidential candidates in the series of Democratic and Republican debates leading up to the Iowa caucuses. What this visualization seems to show is that at this stage the Republican and Democrat candidates are primarily concerned with name-checking the rival candidates within their own parties, rather than attacking the opposition. This is in fact a form of social network visualization that could equally well be applied to a selected set of main characters in a screenplay based on the name-checking of other characters within their dialog blocks in the script. Mrs. Clinton provoked the largest number of name checks indicating that she was perhaps considered the candidate to beat at that

time.



Figure 2.26 - Transcript Analyzer

Jeff Clark's visualization (figure 2.26, see

<http://www.neofornix.com/Projects/TranscriptAnalyzer/index.html> -

accessed 06 July 2008) of the transcript of the Democratic debate of April 16

2008 in Pennsylvania has 3 layers of content, as Clark explains:

The top section shows the distribution of some selected words within the text across a 'timeline', which goes from left to right. Each speech segment is the same width and the height of the small white bars show the number of occurrences of that word for that segment. You can add new words with the text box in the top right corner. You can remove existing words by clicking on them. Right below the word distribution graphs is a similar coloured set showing a spectral decomposition of the text based on who spoke and how much was said. In this case the bar heights give the amount of text for each segment. Click and drag

the mouse left to right to move along the timeline and show the actual text for 3 consecutive segments.

This kind of visualization is also useful for screenplay analysis. The timeline could represent the running order of a series of scenes. The speakers become characters in the script, the text bars representing dialog blocks and the popup text the actual dialog spoken in a given text bar.

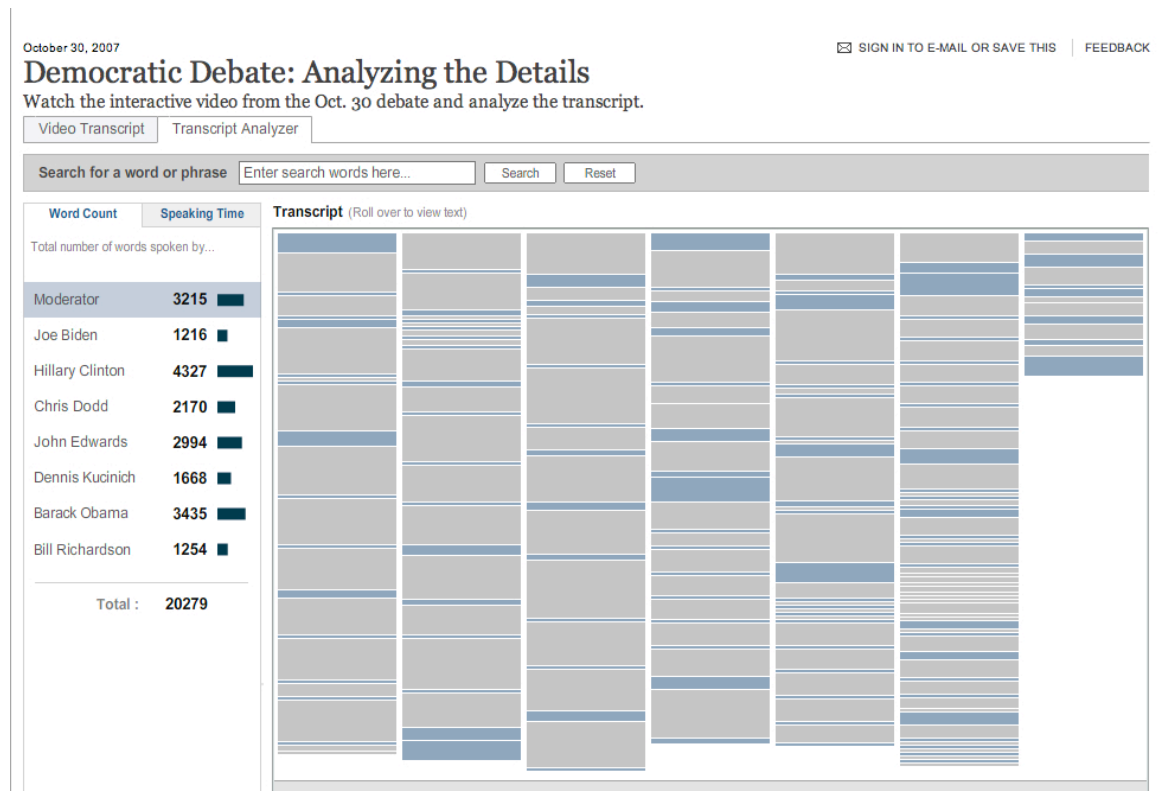


Figure 2.27 – Transcript Analysis

Figure 2.27 (see http://www.nytimes.com/interactive/2008/04/16/us/politics/20080416_DEBATE_GRAPHIC.html# - accessed 6 July 2008) is another *New York Times* visualization from April 16 2008 that provides an analysis of the visualization of the televised democratic debate of October 30th 2007. It shows the word count and speaking times of each participant and pattern of their contributions, the selected speaker's dialog is coloured in blue and replies in grey. Clicking one of the blue/grey blocks reveals the actual text from the transcript. Again this could be used as is on a screenplay as an

indication of the size of a characters part and as a means to navigate the character's dialog throughout the script.

Dialog analysis can also be used to deliver a 'reality-check' on the dialog content of a script by identifying anachronisms that are out of place in the script given its subject matter or time period for example. Benjamin Schmidt, a fellow at the Harvard Cultural Institute, has focused attention on uncovering anachronisms in screenplays such as those by Julian Fellowes for the UK TV series *Downton Abbey*, Tony Kusner for *Lincoln* (2012) and Chris Terrio's *Argo* (2012). Schmidt uses a 'big data' resource, such as Google's Ngrams corpus, to compare the dialog words used in a single screenplay to works from around the same time period in the corpus to identify words or phrases that look out of place such as 'peace talks' in *Lincoln* or 'lifetime achievement award' in *Argo*. Schmidt also makes use of word cloud visualizations to understand how well a script's dialog 'fits' the time period as shown in figure 2.28 for *Argo*.

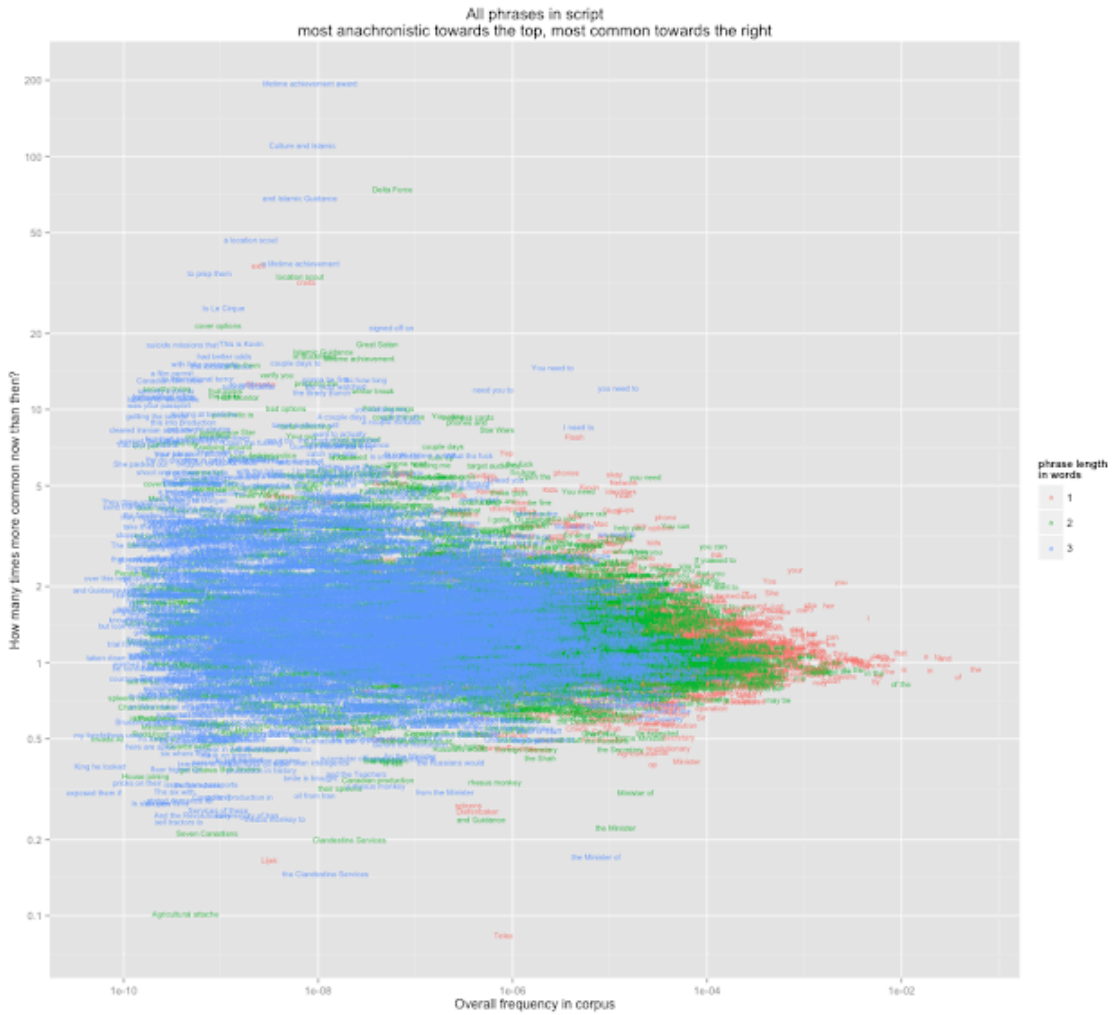


Figure 2.28 – Argo Word Cloud

(see *Nobody Said 'Racial Equality' in 1865: The Anachronistic English of 'Lincoln'* in *The Atlantic* Jan. 10, 2013 and <http://www.prochronism.com/2013/02/current-events-argo-anachronisms.html>, accessed March 2013).

Storyline Analysis

The New York Times also produced storyline visualizations (figures 2.29a-c, see <http://www.nytimes.com/2005/04/24/magazine/24TV.html> - all 3 figures accessed from the online article on 6 July 2008) to compare the relative narrative complexity of selected US TV shows overtime. In the magazine article, *Watching TV Makes You Smarter* (April 24, 2005), Steven

Johnson provides storyline analysis of episodes from US TV shows *Starsky and Hutch*, *Hill Street Blues* and *The Sopranos* to show relative narrative complexity based on the number and depth of the interweaving storylines. Clearly *Starsky and Hutch* is the most simplistic with just one main storyline and a bookending start and finish sub-plot whereas both *Hill Street Blues* has more storylines and *The Sopranos* is the most complex with multiple, denser storylines and considerable overlapping between them.

Figure 2.29a - Starsky and Hutch Storylines

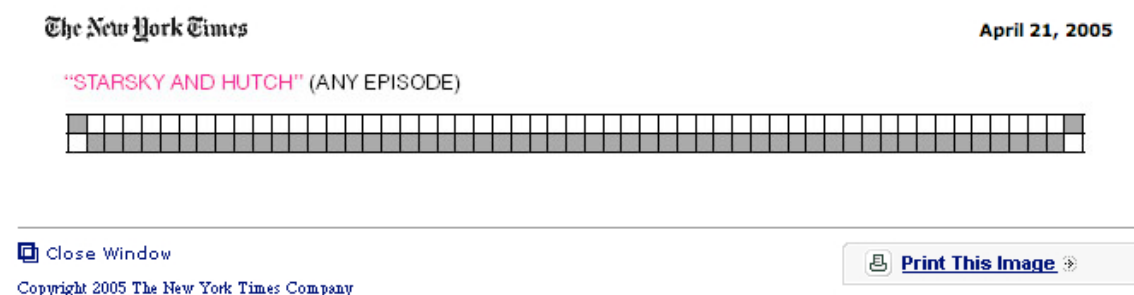


Figure 2.29b - Hill Street Blues Storylines

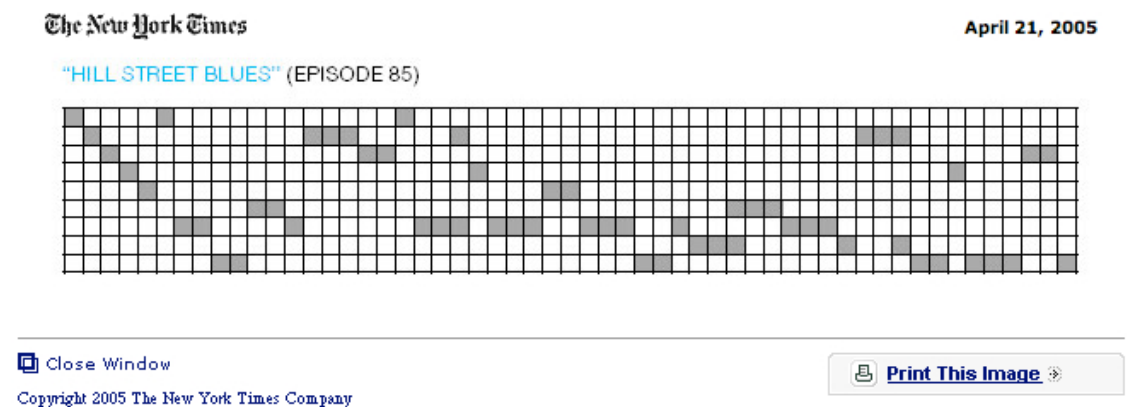
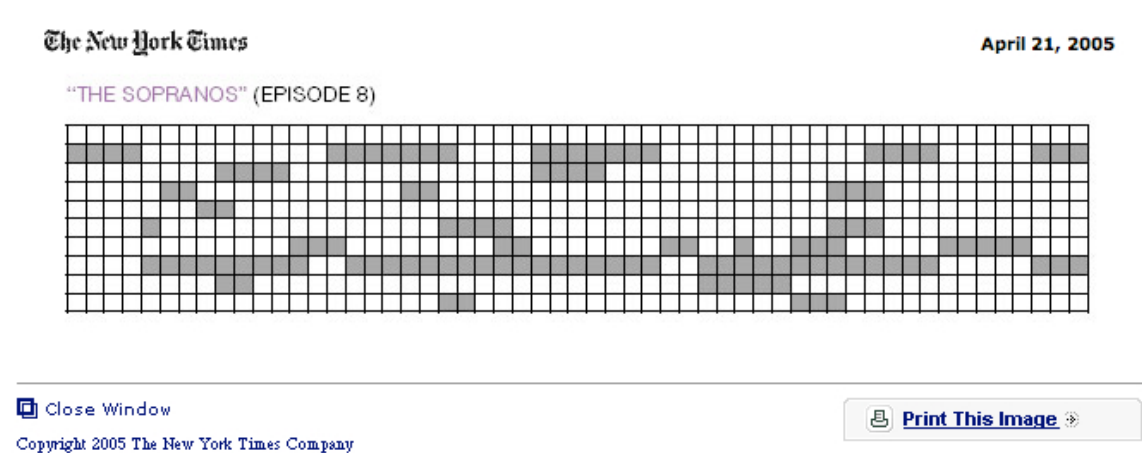


Figure 2.29.c - The Sopranos Storylines



2.8 Using 'Big Data' Tools for Screenplay Analysis

As mentioned in chapter 1, the datafication of the screenplay makes the data available to a new generation of analytic tools designed to work with 'big data' sets and automatically derive certain analytics with minimal user intervention. Here I briefly mention three such tools, all of which were tested with data derived from the Scenepad demo script *Alien* (1979). Because this script was already datafied in Scenepad the dataset needed to feed these analytic tools was sourced from the output of a single query, another reason why datafication of content helps to make analytics easier and increases the range of options available to the analyst.

Splunk

Some rather prosaic, but easy to generate analytics of screenplay data are provided by the Open Source data analysis tool *Splunk* (downloaded from splunk.com – accessed April 28, 2013). To experiment with the kind of analytics that tools targeted at big data problems can deliver, I imported a version of the *Alien* (1979) script into Splunk for analysis.

The core value of Splunk lays in its 'search processing language' – SPL. With the screenplay data in Splunk, I generated some basic 'top n' analyses of the data, which are discussed below.

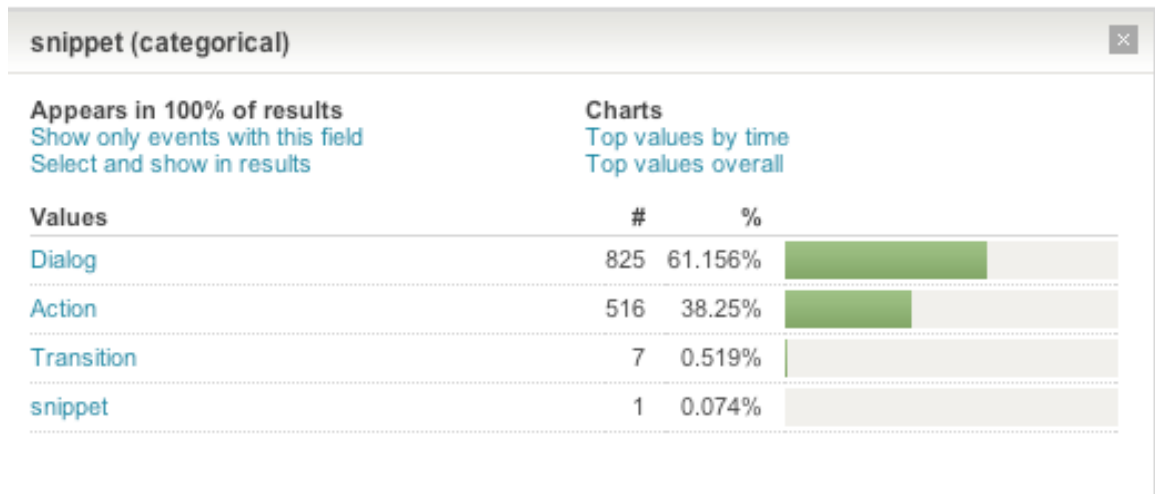


Figure 2.30 – Splunk Snippet Analysis

Here we see that almost two-thirds of the *Alien* script consists of dialog snippets and one-third action snippets. At first glance this might give the impression that the script is dialog-heavy and action-light but in reality this is hardly the case. Dialog content in *Alien* is of far less importance than action content to both the narrative flow and the characters – since much of the action consists of the characters being killed off by the Alien.

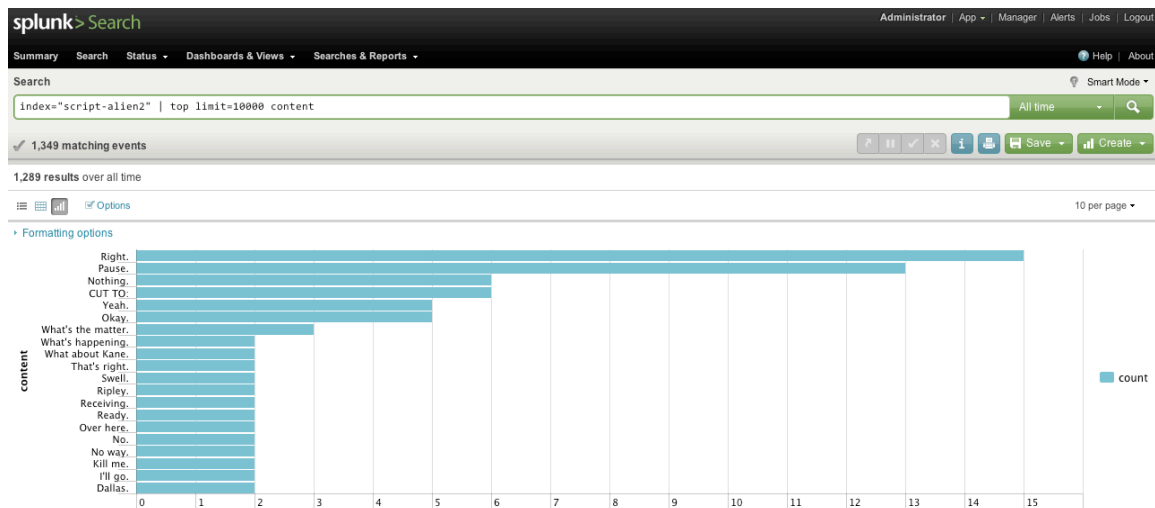


Figure 2.31 – Splunk Word Analysis

Splunk highlights 'right' as the most frequent word in the snippet content (15 occurrences), a word that many might regard as a 'noise' word rather than anything that provides any meaningful analysis value.

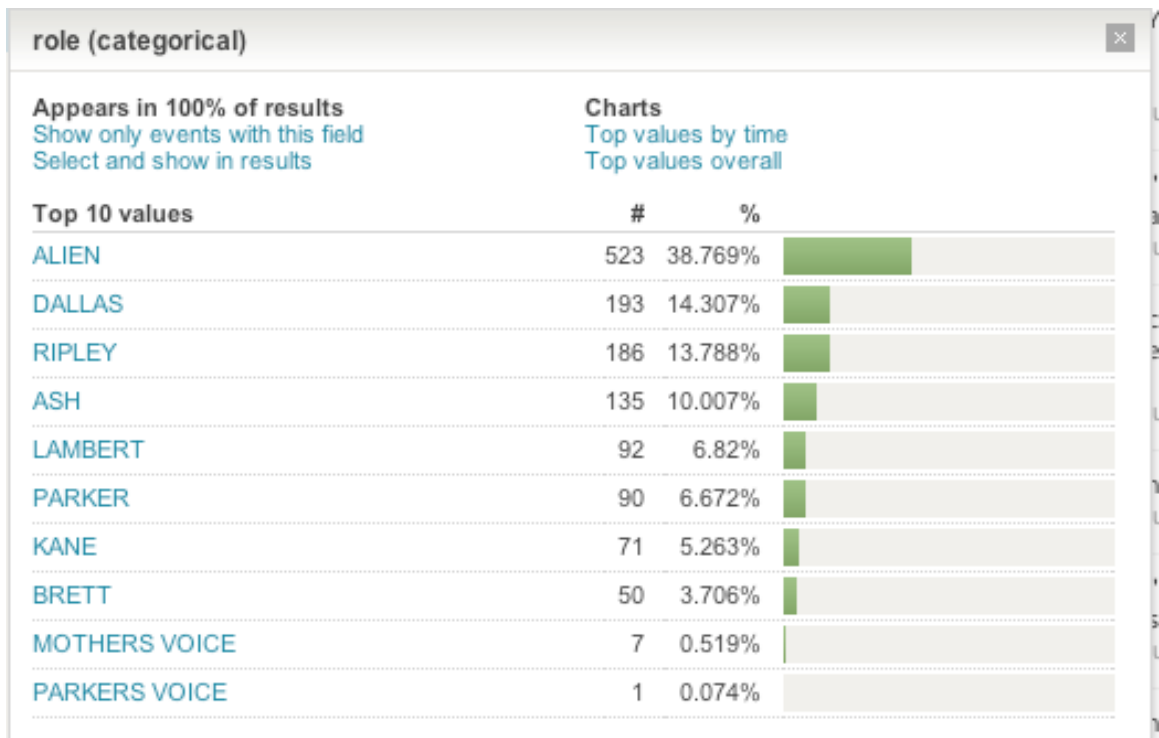


Figure 2.32 – Splunk Role Analysis

Splunk tells us that the 'Alien' role is referenced in 523 snippets. This at least indicates that the Alien – conventionally the 'antagonist' – is probably the most important role in the screenplay with almost 3 times the mentions of the 'protagonist' Ripley. From this it would be realistic to suppose that the 'causality' for this might be that the Alien is in fact the 'driving' character in the screenplay – the villain rather than the hero(ine).

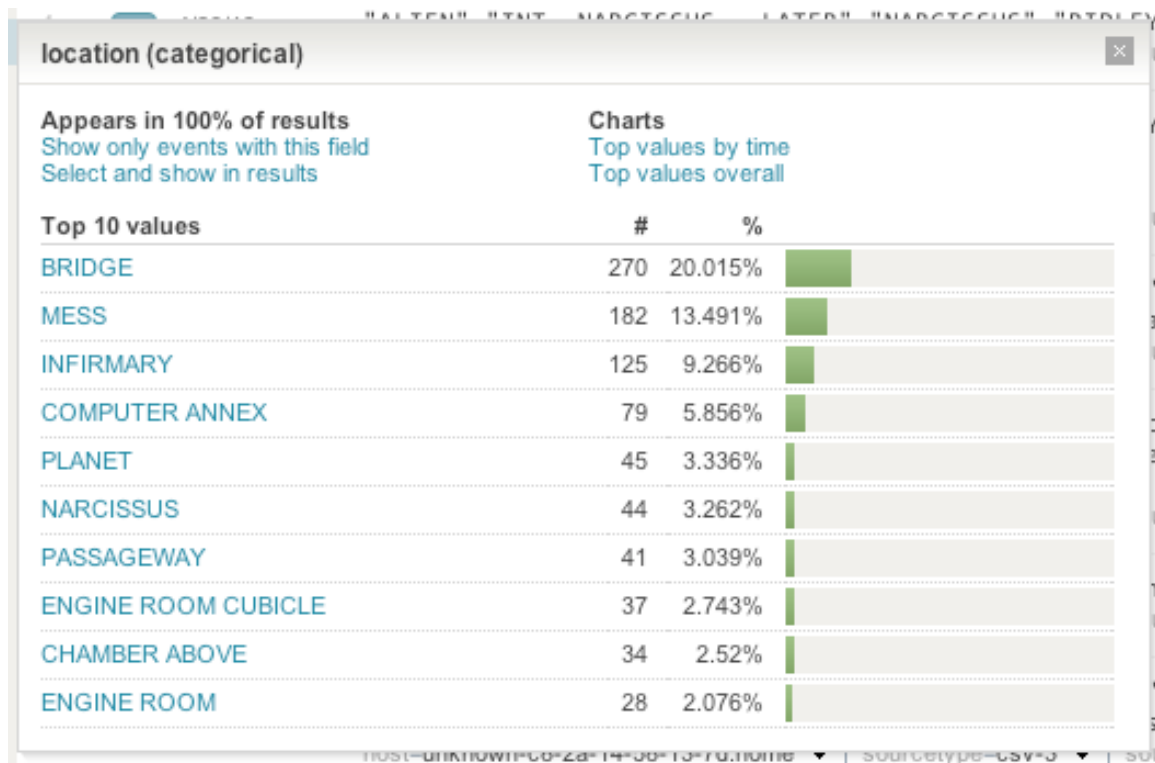


Figure 2.33 – Splunk Location Analysis

Splunk tells us that the ‘bridge’, ‘mess’ and ‘infirmary’ locations are where the most action and dialog snippets take place. This makes sense when you consider that the ‘causality’ for this is that these are three main communal areas on the ship where multiple members of the crew are likely to spend time and to converse with each other.

These simple analyses make minimal use of the SPL and other Splunk features but they are almost ‘one click’ in terms of analytical effort and what they illustrate is that even these simple correlations in the data may be helpful in determining ‘causality’ – i.e. why the data is what it is.

BigML

Compared to Splunk, BigML (see <https://bigml.com>) positions itself as ‘machine learning made easy’ and also provides predictive analytics tools within its analysis toolkit. As with Splunk I loaded in data sourced from the Scenepad demo script *Alien*.

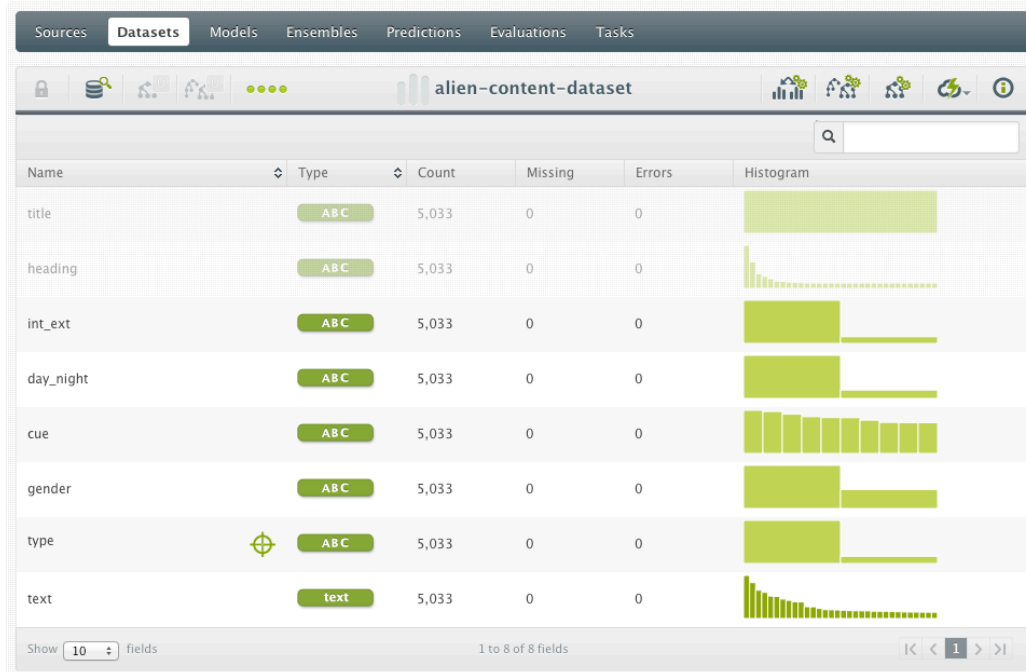


Figure 2.34 – BigML Script Analysis

Figure 2.34 shows the basic ‘immediate’ analysis you get from BigML.

1. The title histogram simply confirms all data is from a single script.
2. The heading histogram shows the distribution of action and dialog ‘instances’ by scene (the top 3 locations are the bridge, the mess and the infirmary).
3. The int/ext histogram shows that most scenes are INT (they take place within the Nostromo).
4. The day/night histogram shows that most scenes are NIGHT (although this is not that helpful as they all take place in space).
5. The cue histogram shows that the top 3 characters in terms of dialog instances are Dallas, Ripley and Ash but also that the instances are fairly well distributed across the cast from Dallas (top) to the Alien (bottom).
6. The gender histogram shows there is just over twice as many instances involving male characters than female (despite a female protagonist).
7. The type histogram shows there are about six times as many action

instances than dialog instances so this likely to be an action-heavy movie.

8. The text histogram shows that Ripley leads the number of text instances (i.e. both action and dialog) over Dallas perhaps reflecting the fact that she lasts longer than he does in the script.

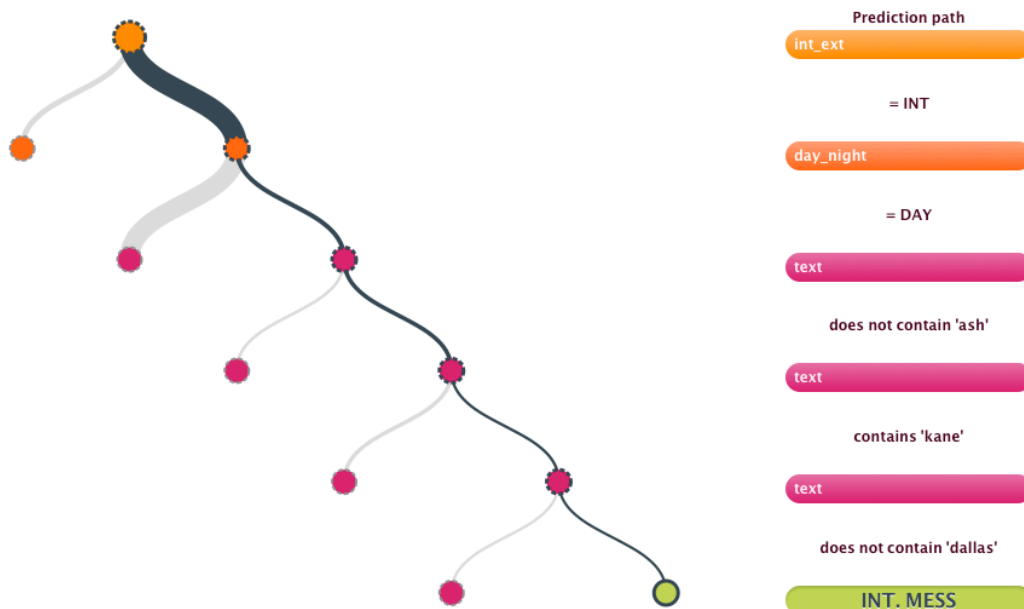


Figure 2.35 – BigML Scene Prediction

Figure 2.35 shows a sample predictive analysis 'path' based on the scene data provided. Essentially all this says is that if the scene is INT and DAY, contains KANE and not ASH or DALLAS it is 93.98% likely to be located in the MESS. It's easy to see how this prediction is largely based on which characters are most often associated with which scene locations (because they speak dialog at that location or take part if action at that location).

Figure 2.36 shows another predictive analysis path using the character cue and a gender attribute to predict that if the cue is not Ripley or Lambert or Mother's Voice then the gender of the character is 99.89% likely to be male. In reality it's 100% likely as these are the only 3 female characters identified in the dataset – all the rest are male. But the point is that BigML is predicting this outcome from the data.

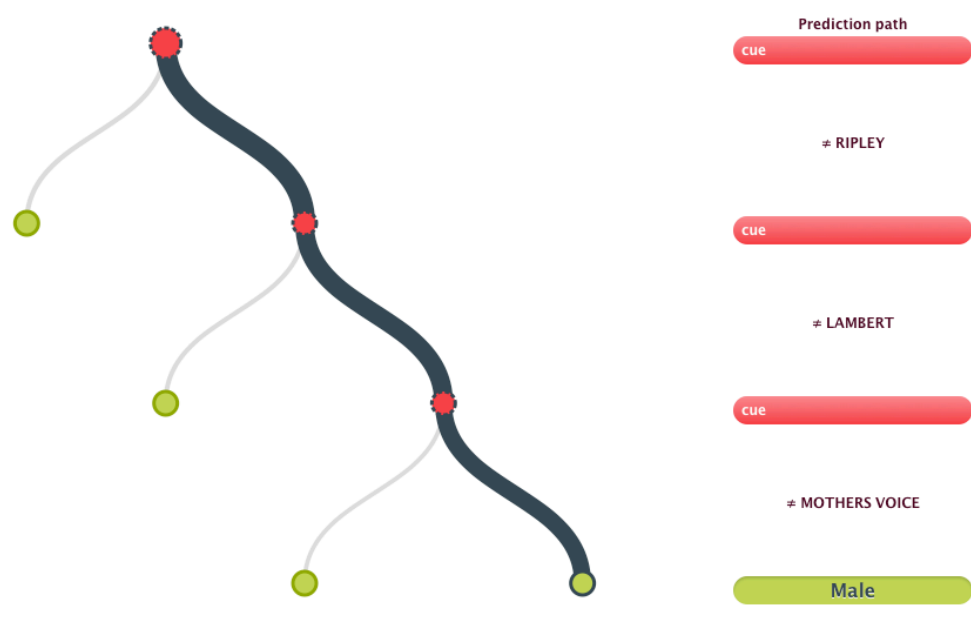


Figure 2.36 – BigML Gender Prediction

Dataseed

Dataseed (<http://getdataseed.com>) is another ‘in beta’ online data analysis application that accepts a file upload and generates some analyses from the dataset immediately after upload. As with BigML I uploaded the Alien screenplay data in a .CSV format and generated the analysis shown in figure 2.37 below. What the dataseed visualization tells us is:

1. There are 1339 action & dialog ‘snippet’ rows in the dataset.
2. 100 snippets are in EXT scenes and 1239 are in INT so we know we are inside most of the time (actually inside the Nostromo).
3. 174 snippets are in DAY scenes and 1165 are in NIGHT scenes (actually the ‘darkness’ of space on board ship).
4. Most of the action and dialog takes place on the Bridge, the Mess or the Infirmary.
5. There are 514 action snippets and 825 dialog (so is it an action-heavy or dialog-heavy script – we don’t know.).
6. RIPLEY and DALLAS have the most important roles (so one or other is likely to be/should be the Protagonist).

7. Males dominate the dialog but female roles have a good deal of screen time (540 vs. 285 dialog snippets respectively) especially considering there are only 2 of them.
8. The Protagonist role has a significant number of dialog snippets (186 vs. 631 for all the other 'Main' characters – indicating significant screen time).

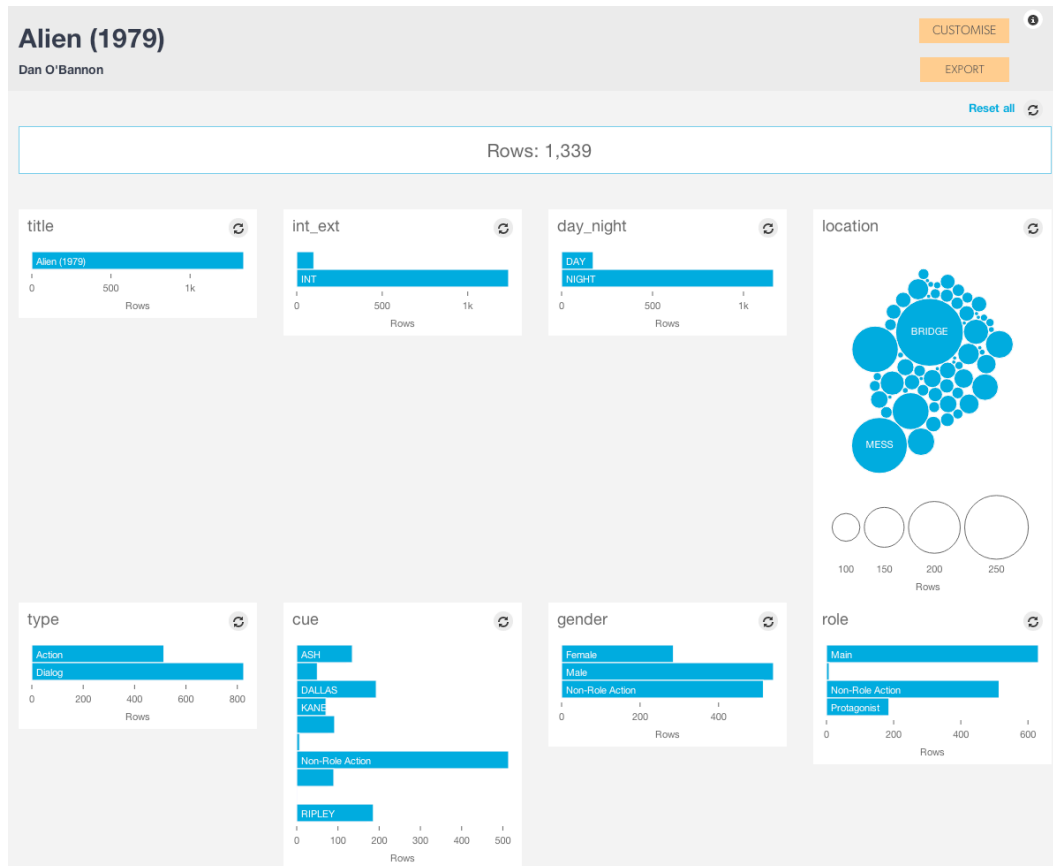


Figure 2.37 – Dataseed Analysis

What these three simple 'experiments' in uploading screenplay data to data analytics tools tells us is that you don't even need a 'Screenwriting 2.0' application like Scenepad with built-in analytics, you just need an easy way to get screenplay content delivered in the form of a data file (as opposed to a document text file) to enable these kinds of tools to deliver a 'starter' set of analytic visualizations from the content.

The data file provided to both BigML and Dataseed was a simple .CSV file like the sample shown in figure 2.38 below.

	A	B	C	D	E	F	G	H	I	J	K	L
1	title	scene_id	sceneseq	int_ext	day_night	location	seq	type	text	cue	gender	role
2	Alien (1979)	5	1	INT	NIGHT	INFIRMARY	4	Action	Distressed iv	Non-Role Act	Non-Role Act	Non-Role Action
3	Alien (1979)	2	2	INT	NIGHT	ENGINE CUB	1	Action	Circular, jam	Non-Role Act	Non-Role Act	Non-Role Action
4	Alien (1979)	7	3	INT	NIGHT	BRIDGE	6	Action	Vacant. Two	Non-Role Act	Non-Role Act	Non-Role Action
5	Alien (1979)	7	3	INT	NIGHT	BRIDGE	7	Action	b.g	Non-Role Act	Non-Role Act	Non-Role Action
6	Alien (1979)	7	3	INT	NIGHT	BRIDGE	8	Action	. Electronic h	Non-Role Act	Non-Role Act	Non-Role Action
7	Alien (1979)	3	4	INT	NIGHT	OILY CORRID	2	Action	Long, dark. E	Non-Role Act	Non-Role Act	Non-Role Action
8	Alien (1979)	4	5	INT	NIGHT	CORRIDOR	3	Action	Long, empty.	Non-Role Act	Non-Role Act	Non-Role Action
9	Alien (1979)	8	8	INT	NIGHT	CORRIDOR T	9	Action	Lights come	Non-Role Act	Non-Role Act	Non-Role Action
10	Alien (1979)	9	9	INT	NIGHT	HYPERSLEEP	10	Action	Explosion of	Non-Role Act	Non-Role Act	Non-Role Action
11	Alien (1979)	10	10	INT	NIGHT	GALLEY	11	Action	Kane plugs ir	Non-Role Act	Non-Role Act	Non-Role Action
12	Alien (1979)	10	10	INT	NIGHT	GALLEY	12	Dialog	Rise and shin	KANE	Male	Main
13	Alien (1979)	11	11	INT	NIGHT	HYPERSLEEP	13	Action	Another lid p	Non-Role Act	Non-Role Act	Non-Role Action
14	Alien (1979)	11	11	INT	NIGHT	HYPERSLEEP	14	Dialog	What time is	LAMBERT	Female	Main
15	Alien (1979)	11	11	INT	NIGHT	HYPERSLEEP	15	Dialog	What do you	KANE	Male	Main

Figure 2.38 – Sample of Uploaded Data File

Summary

Screenplay analytics, the analysis and visualization of screenplay content, is a very underdeveloped area, both for individual screenplay analytics and for screenplay corpus (e.g. genre) analytics. Screenwriting ‘how-to’ manual writers, media publications and even the odd academic are among the few sources for a small number of screenplay content visualizations derived from analysis of the screenplay text. Some generic data analysis tools can be used to generate analyses and visualizations of screenplay content, once it has been ‘datafied’ into an appropriate format for use by the tool. Despite the lack of screenplay analytic capabilities in most screenwriting applications, screenplay content can be easily imported into generic data analysis tools like *Splunk* in order to drive some level of analytics from the text.

Much of this lack of screenplay analytic activity is a result of the prevailing orthodoxy of working with screenplays as textual documents rather than screenplays as data. As more screenwriting software datafies the screenplay text, and especially if structured and standardized formats are used, it is very likely that new kinds of analysis and visualization of screenplay content will emerge, as applying analysis and visualization to the ‘datafied’ text will be easier to do and the text will become more easily accessible to a wider range of analytical and visualization software.

But screenwriting 2.0 is about more than screenplays as data and leveraging that data for content analysis and visualization. It is also about the screenplay as social network and collaborative business process and about new perspectives of the way screenplays are made or screenplay as design. These are the topics discussed in my next chapter.

3 Screenplay as Social Network

Writing and realizing a screenplay as an audio-visual work is not a discrete event but a creative and potentially business process that often reflects a long-duration lifecycle, sometimes extending across many years, with many different activities involving many different stakeholder roles as participants in the process. Therefore screenwriting implies a social process with the screenplay text itself providing the context or 'anchor' for a social network to collaborate around.

A single writer working on a 'closed' screenplay (i.e. stored on their local PC and not shared with others) is not a social network but a writing partnership with just one other partner corresponds to the minimum dyadic relationship needed for the basis of a social network, especially if this relationship includes use of an 'open' screenplay that is stored online and accessible via the Web for sharing with other collaborators.

Although a screenplay shared with a large group might benefit from the 'collective intelligence' gained from what James Surowiecki popularized as 'The Wisdom of Crowds' in his 2005 book of the same name, it is much more likely to benefit from exposure to a more finite and smaller group of collaborators where, according to Kao and Couzin (2014):

...it is the noise inherent in these small groups that enhances their accuracy, allowing individuals in such groups to avoid the detrimental effects of correlated information while exploiting the benefits of collective decision-making.

A simple example of an early form of online social network is the blog. Typically, the owner/producer posts news, reviews or opinions and the reader/consumer posts comments, rates the content or reblogs the content to co-create a conversation or debate around the post topic. Here, the social network paradigm we are more interested in is that of the online social networking site/application (SNS/SNA), such as Facebook or Twitter, primarily designed to function as social networks that encourage the

crowdsourcing of user generated content (UGC) that typically takes the form of text, images or audio/video clips. I refer to social network applications (SNAs) below, not social network sites (SNS), since in my view the term 'site' implies primarily a content-delivery purpose whereas the term 'application' implies both delivering content and the functionality needed to manage that content.

According to boyd and Ellison (2007), who trace SNS development over the decade 1997-2007, social network sites are defined as:

...web-based services that allow individuals to (1) construct a public or semi-public profile within a bounded system, (2) articulate a list of other users with whom they share a connection, and (3) view and traverse their list of connections and those made by others within the system.

However this is a rather technical definition focused on online SNSs. For our purposes here, I define a social network as comprising actors with relationships between them. In this context those relationships may be developed and managed both face-to-face and online. The exchange of data between participants engaged in a social network creates social capital. The relative engagement of the network actors is what enriches this social capital. This engagement is reflected in the amount of UGC aggregated around a specific actor and their individual content items (e.g. blog posts, Facebook pokes, Twitter tweets). This UGC can take various forms including commenting or rating (e.g. 1-5 star ratings or Facebook likes) or referencing (e.g. blog backlinks or Twitter retweets). The social network is anchored by a focal point around which a specific actor's social capital revolves.

SNA	Actors	Anchor	Items	Engagement
Facebook	Friends	Wall	Pokes	Likes

Twitter	Followers	Timeline	Tweets	Retweets
Wordpress	Contributors	Blog	Posts	Comments

Table 3.1 – Social Network Characteristics

Another way to look at a social network is as an instance of a creative system model as proposed by J T Velikovsky’s blog post, *CREATIVE PRACTICE THEORY: The Model*, of 15, Dec 2012 (see <http://storyality.wordpress.com/creative-practice-theory/> - accessed 09 July 2013)

that identifies some characteristics of a systems model that could equally be applied to a social network: The ‘domain’ (e.g. screenwriting), the ‘field’ (e.g. screenwriters) and the ‘agents’ (e.g. individual writers) and arguably the ‘symbolic capital’ at the centre of all three is the screenplay content itself.

In fact, some screenplays could involve multiple social networks that overlap each other to reflect the different domains, fields and agents that are involved with the script over its process lifecycle as shown in figure 3.1 below:

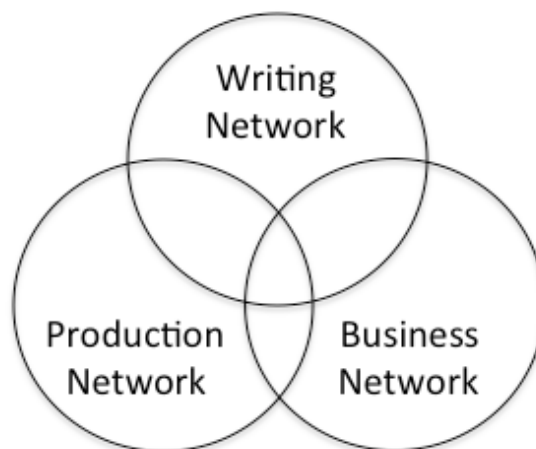


Figure 3.1 - The 3 principal screenplay-anchored social networks

The writing network involves the writers, readers/analysts and others who are involved in writing, rewriting and 'polishing' the script content. The business network involves those agents, producers and others involved in financing, selling and marketing the screenplay. The production network involves those involved in converting the content into a movie including the director, talent, cinematographers, production managers and others. In each case the 'domain' and 'field' is different, the 'agents' reflect a different stakeholder group and the 'symbolic capital', while fundamentally the screenplay content, is changing all the time and presented in different ways depending on the network using it (e.g. production shooting script vs. business selling treatment). What this triple domain perspective indicates is that the screenplay can be at the heart of a rich social network.

In this chapter I will discuss the functionality expected of an online social network, the range of screenplay stakeholders and screenwriting as a process.

3.1 Functionality of Online Social Networking

We recognize an online SNA by its provision of specific functionality that promotes social networking and fosters ongoing engagement with the other network participants and their content.

Typically, SNAs operate in both public and private modes in terms of their content visibility. The ability to flag content as public or private (either to you or a specific group of friends or followers within your social network) is integral to most SNAs. However a SNA anchored by a screenplay that is commercially 'in-play' is more likely to operate as a 'closed' network where the user community and access to the content is prescribed by the screenplay owner or copyright holder. Participants in the network must be invited to join so it is not 'open' in the sense that any online user can view the screenplay content and interact with it (as is the case with a typical blog

for example).

A SNA is by definition multi-user. Note that many screenwriting applications are not since they often assume a single user working with the screenplay content – namely the owner-writer. Multi user access also means that many SNAs recognize the need for user roles (e.g. editor, author, viewer) to effectively manage content access and usage rights. However single user screenwriting applications do not support user roles since they are not necessary.

In general, most online SNAs support the functionality outlined below:

Create and grow their network by sharing content

In an 'open' social network the aim is to expand an individuals network as widely as possible by making it easy to share content between users. On Facebook this is known as 'friending', on Twitter 'following' and on LinkedIn 'linking'. Sharing activities such as re-tweeting, re-blogging etc. allow specific tweets or blog posts to go 'viral' very quickly, reaching many more participants in the social network, if the content happens to catch the imagination of the crowd. This is the power behind tweets that 'trend' on Twitter or YouTube videos that record millions of views.

Contribute content

All social networks depend on user-generated content (UGC) to function: Facebook wall pokes and Twitter tweet posts are both examples of UGC, as are photos posted to Flickr or video clips posted to YouTube, and it is this UGC that comprises much of the core 'data' of the online social network (the rest being content added by other users specifically intended to enrich that core content). In most cases the initial UGC (blog post, tweet, photo etc.) is itself the trigger for generating new UGC around it. It is this content enrichment process that creates the 'conversation' that in turn has the potential to add value to the original content.

Rate content

In a social network, rating content is a way of crowdsourcing an indication of the quality of that content and also a means for filtering network content programmatically so that 'cream rises to the top' in the form of an automated ranking based on user ratings. The Facebook 'like' is a simple rating system for ranking leading and lagging content. The Twitter 'retweet' is another way of saying you value content by actively sharing it. The most liked and most retweeted content is likely to indicate popularity that is perhaps a reflection of content quality.

Crowdsourced ratings are highly effective in the e-commerce domain, for example the 1-5 star rating system for both products and sellers on Amazon that may have a direct impact on the buying decision of a customer. Usually anyone can rate content and it is up to content consumers to decide how relevant/realistic those ratings are to them individually.

Comment on Content

Providing feedback in the form of comments is an integral part of every social network and a simple example of a peer review process in action. Commenting on content is a key way to encourage the original author to review and revise their content based on the multiple perspectives reflected in a comment stream. Comments are generally viewed in context i.e. alongside or below the content being commented on. The number and quality of comments is generally considered to be an indication of the original 'stimulating' content's quality.

A problem with many SNA comment streams is that they contain a great deal of 'chaff' e.g. of the 'awesome dude' or 'this sucks' variety. This chaff is seldom very helpful to the content creator. That is why it can help to force comment senders to categorize their comments, for example as 'suggestions' or 'problems', which in turn helps the comment receiver to better analyze these comments and take action. Categorizing comments will not prevent

chaff but may help to reduce it.

Aggregate content from multiple sources

Content aggregation becomes important when there are a multitude of places that content can be posted to or sourced from. This has become more important given the pervasive adoption of mobile devices, especially smartphones, that can capture content anywhere, anytime and then post it to various online SNAs. Content aggregation will become more important as more devices become part of the Internet-of-Things and can communicate data wirelessly to a specific collection application. For example when a camera used to shoot a script in production, a 'thing', can automatically communicate metadata about the filming via the Internet to a screenwriting application where it can be stored along with other scene data and metadata.

Early forms of online content aggregation included online newsreaders that aggregated RSS (Really Simple Syndication) feeds from multiple source sites. The use of RSS in this way facilitated easy syndication of content to multiple content consumer target applications from a single publisher source application.

Other than via RSS, the usual way to share and aggregate content programmatically by use of the source and/or target application's API (Application Programming Interface). An API generally provides a secure way for a consuming application to request and get specific content from a publishing application. All leading SNAs – for example Twitter and Tumblr - offer such APIs to promote posting and querying their SNA data programmatically.

Activity Updates and Alerts

A SNA highlights the engagement of its users by making much of their engagement activity transparent to other users in the form of activity streams and alerts (e.g. 'X posted Y' or 'A is now linked to B'). An activity

stream visualizes this activity in the form of an activity list or a posting timeline that constantly updates to reflect the latest user activity. Alerts may be triggered to alert specific users to specific activity based on a set of alerting preferences managed by the user. The alert ('C likes your D') is usually delivered automatically to the interested users by email or SMS text message. The surfacing of activity streams and automated alerting help to increase engagement with the SNA by prompting users to respond in some way – even if only to login to the SNA rather than to elicit any specific, formal response.

Gamification and PBLs

Points, Badges and Leaderboards (PBLs) have also become an aspect of social networks, borrowed from computer gaming. According to Werbach and Hunter (2012), 'gamification' of social networks can be a way to increase engagement with and feedback from a social network. This gamification often uses PBLs by awarding points for activities such as posting content comments or rating 'likes', leading to award of badges as participants 'level-up' to pre-defined award levels. High scoring participants are then showcased on leaderboards e.g. Top 10 Commentators (on the network) to provide recognition for their efforts. Gamification and PBLs probably make less sense in a relatively small social network focused on a single screenplay's content but might make sense in a larger network focused on a corpus of screenplays for a use case such as genre analysis and are certainly more applicable to sites focused on delivery or discussion of movies for example, rather than screenplays

3.2 Screenplay Stakeholders

Most of the current generation of screenwriting applications does not support this social networking perspective of screenwriting. This is largely because many were conceived or developed before the rise of social networking as an Internet phenomenon and so tend to focus on screenwriting as an individual creative process rather than as a

collaborative business process that eventually can involve a social network of both creative and commercial stakeholders.

That professional screenwriting and script development is a business process is well understood outside the world of screenwriting software. For example, the Creative Skillset's overview of the script department (<http://www.skillset.org/film/jobs/script/> - accessed 04 September 2012) states:

Although the Screenwriter or Screenwriters are central to the script writing process, like every other aspect of film making, development is collaborative work and typically requires the creative input of a number of other film professionals including: Script Readers, Writer's Agents, Producers, Development Executives, Script Editors, and, eventually, Directors, who are often involved in the final versions of the screenplay and shooting script. When projects are developed as part of a Screen Agency, Broadcaster, or National Lottery funded film scheme, Development Executives from these agencies and script development staff may also be involved in the process.

This process reflects the journey that the script makes from an initial idea to the final shooting script from which a movie is more or less directly shot.

In fact screenwriting as a collaborative activity has much in common with the idea management activity within the business domain of innovation management. The progression of the screenplay development process from ideation to realization has many steps that include genuine commercial 'stage-gates' at which business decisions are made – optioning, developing, green-lighting - just like any other commercial product development process. And like the manufacture of a physical object, a screenplay can generate many versions and prototypes in the form of script drafts and treatments, storyboards and pre-viz visualizations.

A script may begin its journey owned by an individual screenwriter or a partnership or team of screenwriters. The former is a common paradigm for either a speculative or commissioned feature film screenplay and the latter for a screenplay destined for TV – especially if written as an episode within an established series leveraging an established writing team with a

showrunner and a series bible.

In terms of collaborative screenwriting, what is of interest here are the similar yet different workflows the individual and team scripts follow, the common phases within these workflows and the needs and activities of the various roles who participate in the workflows. In practice, the original or 'initiating' screenwriter may not in fact participate in the process from start to finish. This originator may in fact be 'cut loose' from the moment the script copyright ownership is purchased, replaced by other writers while the script is 'in development' or simply marginalized by more powerful production or commercial roles such as the Director or Producer even if they remain on board for the duration of the project.

Screenplays are either commissioned (i.e. paid-for by a third party) or produced speculatively (i.e. without advance payment) by a screenwriter who hopes to sell his work. Kawin (1992:302-326) provides a thorough explanation of these two paths of screenplay genesis. From the work of MacDonald (2010) on the screen idea and its development, and anecdotal descriptions of working with a writing partner (e.g. <http://spitball.com/2009/04/how-we-collaborate-on-a-screenplay>, accessed 28 August 2012) it is clear that the original author(s), or any other writer involved with the script, is unlikely to be individually largely responsible for or in control of every draft of a script from pre to post production (unless the screenplay is created and produced by a writer/director). In other words most, if not all, screenplays are co-created that are eventually realized cinematically.

As a collaborative process it may be that the script is initially written and/or developed by a writing partnership or team – this is often the case with scripts written for television (as the classic US television series *The Dick Van Dyke Show* and more recently *30 Rock* portray). But whatever the genesis of the script, the process of writing and rewriting, combined with the way that scripts are marketed, developed and produced will almost certainly mean that any 'final' shooting script will reflect input from a wide range of

stakeholders in addition to the original screenwriter, who may in fact have been removed from the project a long time ago.

There are a number of stakeholder roles involved in the screenwriting process. Here I identify some key roles, their involvement in the process and their potential interest in being able to analyze the screenplay content (as facilitated by the datafication of the content).

Writers

The writers include anyone who is involved in writing or rewriting/revising screenplays. The original author(s) and subsequent 'rewriters' and 'polishers' attached to the script are included in this role. This is the primary role involved in conceiving and writing the initial screenplay content (data and metadata) re-writing it and being officially credited with the writing. This role is motivated by the desire to write a better script to increase its commercial viability or to respond to the needs of the producer/director who may be subject to their own budgetary or artistic forces.

Investors/Producers/Directors

Investors, Producers and Directors include anyone who is involved in participating in the financing, buying or optioning of a screenplay and who may require the work to be formally evaluated prior to any investment or purchasing decision. This group is motivated by the need to identify scripts that are judged likely to succeed commercially and therefore worth investing in for an acceptable financial return.

Script Readers/Story Analysts

Script readers and story analysts include anyone who professionally (i.e. as a job role or as freelancer for a fee) reads and analyzes screenplays. This group often works on behalf of investors/producers/directors or writers, in

order to produce ‘coverage’ or any other kind of report or analysis that evaluates the artistic merit and/or commercial potential of a screenplay. These roles, like Agents, often act as ‘gatekeepers’ in terms of their influence on the continued progression of script into the production or realization process.

Production Managers

Production managers include anyone who is involved in managing the production of a movie from a screenplay. This group is interested in the implications of a screenplay from a production perspective, for example the script editor for continuity or specialist crew and managers focused on locations, props, makeup, costumes or special effects. The screenplay acts as the primary source for the production breakdown reporting, shot lists and daily call sheets that are essential to the mechanics of movie production.

Talent

Talent refers to the actors who are potentially or contractually attached to a project. Their motivation for engaging with the screenplay content may be in better understanding the screenplay to perform their role or for deciding whether or not to buy or option the screenplay for their own production companies or whether to attach themselves to an existing production project because of the quality of the role or its ‘Oscar’ potential for example.

Agents

Agents sit between screenwriters and producers/directors/talent to perform a “filtering” and selling role that may also require the engagement of script readers/analysts to pre-qualify screenplays before the agent can pitch them to potential investors in the script (e.g. producers/directors/talent/angels). They may be interested in using screenplay content analysis and visualization to cut their slush pile of

unsolicited “spec” scripts from unknown writers or to help them apply additional due diligence to solicited scripts or those from existing clients.

CGI/SFX/VFX

The increasing use of green/blue screen techniques and computer generated images (CGI) to create special and virtual effects (SFX/VFX) in movie production means that a relatively recent addition to the stakeholder group are the technicians/programmers and others required to deliver these effects. Here the script functions as a functional input into the software development project required to realize this aspect of the movie production.

This range of roles indicates that individual screenplay data ‘consumers’ will need to interact differently with a screenplay at different stages of the script development and production process depending on their point-of-view.

Note that here, we are not concerned with the audience of the finished movie as a screenplay stakeholder. The audience of a finished movie is seldom aware of or interested in the screenplay after it has changed form i.e. from written artefact to produced movie. Nor are other stakeholders involved the post-production sale and distribution of the movie relevant here.

3.3 Screenwriting as Business Process

Screenwriting as a business process can be discussed in at least three ways:

1. As having pre-production and production phases
2. As having pre-sale and post-sale phases
3. As an idea development process

Again, here we are not concerned primarily with the post-production business process, when the screenplay has changed form (from text to movie). Once a movie is made, the screenplay itself has changed form from a

fluid, 'living' document of a potential movie to a 'frozen' transcript of a realized movie. We are only concerned with the business process that relates directly to the development of the screenplay as potential movie.

Pre-Production

Pre-production is the process phase that occurs before the actual production (or shooting) of a movie begins. At this point the script may not even have been written, if commissioned by a production company or not bought but only optioned, if a speculative script. The stakeholders involved in pre-production could include:

- The Screenwriter(s) primarily responsible for the initial script draft(s).
- The Script Analyst(s) who may help the screenwriter(s) to refine the script by providing analysis or coverage of the script.
- The Script Reader(s) responsible for assessing the commercial worth of a script in terms of its production potential for a specific production company.
- The Script Rewriter(s) and Polisher(s) who may be hired to rewrite or otherwise improve specific aspects of the script pre-production.
- Producers, Directors and/or the Talent who may provide input resulting from their reading and interpretation of the script once they become 'attached' to a project.

At this stage, the content of the script can be highly fluid since it does not yet have any direct production (i.e. financial) impact.

Production

Production is the process phase in which shooting the movie takes place and the bulk of the movie budget is spent. At this point, a substantive script will normally exist (unless improvisational techniques are being used by the director and the script is being created largely 'on-the-fly' on set) and the

production company will have bought the rights to the script from the original screenwriter and assumed ownership. The stakeholders involved during production include:

- The original screenwriter (but only if their contract with the production company continues their involvement).
- The Producer(s) who may influence the script to meet production timing/budgeting demands.
- The Director(s) who may influence the script to reflect their personal directorial "vision".
- The Talent who may influence the script to reflect their personality or acting preferences.
- Various production managers, who may influence the script, say to make it easier or more cost-effective to make into a movie from their production perspective.
- The Script Doctors(s) and Polisher(s), the 'hired guns' who rewrite or otherwise improve the script content as required during the shoot.

The screenplay content may change significantly during this development process but any revisions are usually clearly identified, not only for writing credit purposes, but also as they may have a production/financial impact. Traditionally, at this stage, revisions are added to the script on different coloured paper to reflect different revision cycles. These revisions may include line-by-line edits or wholly new scenes or cause whole scenes to be deleted altogether. Characters and locations, special effects and sounds may also be developed, dropped or added. Dialog in particular may be substantially revised as it is successively polished for optimum impact. Locations may be dropped and action scaled back to reflect budgetary concerns.

Of course, it is possible for a script to change post-production during the editing process - say to dub on a new piece of dialog – and clearly an editor who cuts a movie is in effect cutting the shooting script content. But by the time a movie is in post-production, the original screenplay is essentially a historical document and it is the transcription of the final movie that becomes the truly representative and actual ‘final draft’ screenplay version.

Pre and Post Sale

Another way of viewing the screenwriting process, specifically from the screenwriter’s perspective, is to divide it into pre-sale and post-sale phases. As the discussion above indicates, a pre-production draft may bear little resemblance both in form and content to the final shooting script. In fact they are being used for different purposes. The purpose of a pre-production script is to sell the story vision of the writer, whereas the purpose of the production shooting script is to efficiently convert the script into shots that can be edited into a movie.

During this selling process, the screenplay also is represented by a variety of documents that have different purposes within the process: Treatments, synopses, contracts, coverage reports, ‘the first 10 pages’ and so on that reflect the fact that not every stakeholder involved in this process is interested in the actual screenplay content itself but only a version of it, appropriate to their needs as a stakeholder. The screenplay itself still functions as the ‘anchor’ for all these spin-off documents but it does not stand-alone it is part of a document-set.

Pre-sale, the script has latent rather than actual value and the script is essentially totally controlled and owned by the original writer(s). Pre-sale, the job of the screenwriter is to improve the script so that its latent value can be realized, resulting in either the script being optioned or bought or at least winning some kind of award or competition that may help to improve its chances of being optioned or bought.

The way this is done is by an ongoing cycle of revisions and rewrites, as I outline in figures 3.2a & 3.2b, based on the writer's own reflection or internal feedback combined with coverage from readers and analysts, and more or less external feedback from co-writers, and from pitches to festivals, competitions, production companies and other interested parties. Much of this cycle involves significant reading and re-reading of the script, which is both time-consuming and repetitive.

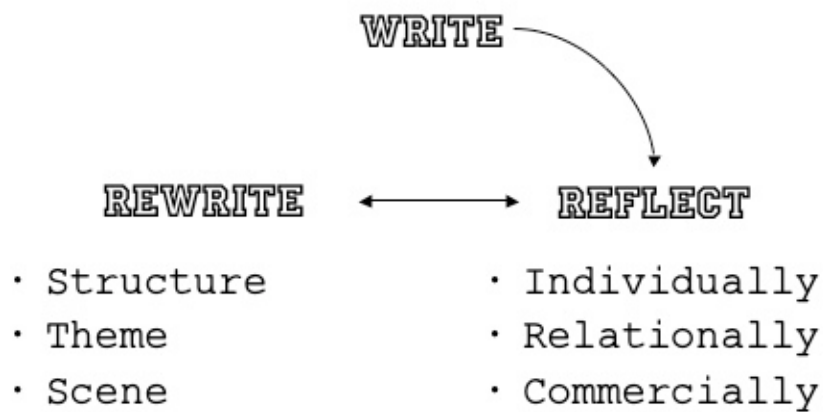


Figure 3.2a - Write – Reflect - Rewrite

Production company readers and paid-for script analysts generally produce a formal written report that summarizes their personal analysis of the quality of the screenplay content and its commercial viability. Good coverage is valuable feedback to help a writer to improve the script or to facilitate the marketing of their screenplay (for example a script that has coverage from a bona-fide production company is by definition 'in play' – whatever the perceived positive or negative nature of the coverage). A useful summary of the content and purpose of a coverage report can be found in Garfinkel (2007:53-95).

Coverage is probably the only formal script 'analysis' that a screenplay will ever get pre-sale. But all coverage inevitably reflects the opinion of an individual script reader or analyst– albeit an individual likely to be skilled in providing this kind of analysis. For this reason, any analysis is inevitably

broadly subjective based on intuition and market experience rather than broadly objective based on the evidence of the script data. Subjectivity may be a good thing, since it reflects certain requirements and biases prevalent in the industry at the time, or a bad thing, since it reflects certain reader-specific or production company prejudices or preferences that may ignore genuine qualities in the screenplay content.

Adding a new step into the screenplay re-writing process, that of some kind of script content analysis and visualization, has the potential to significantly enrich the process as a whole by creating useful new feedback loops, as figure 3.2b indicates.

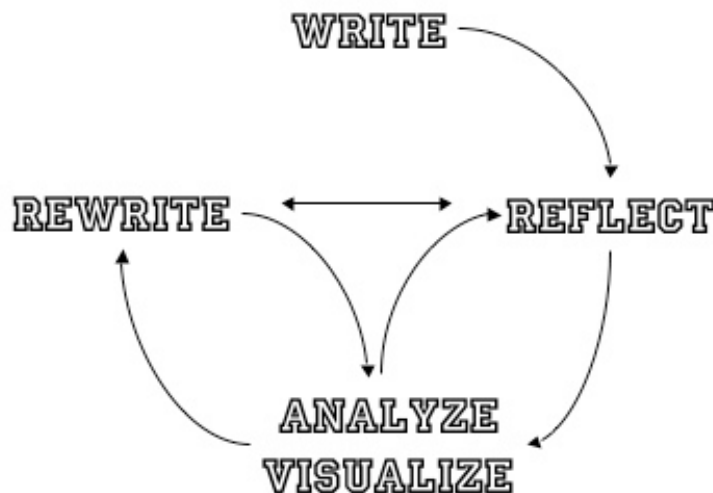


Figure 3.2b - The Enriched Screenwriting Process

However, it is unclear what constitutes a 'high quality' script, pre-production. For example does this mean a script that is judged as having a satisfying visual narrative or one that is judged as having the potential for high box-office (i.e. commercial), a potential Oscar winner or what?

Clearly, from a commercial perspective, this judgment depends largely on what a particular buyer is looking for in order to satisfy the needs of the market they are targeting the movie at. It may also be partly determined by the availability of specific funding streams or tax breaks to defray

production costs. Whereas from the screenwriter's perspective it may depend on what kind of screenplay the writer set out to write. Is the writer satisfied with what they have created in creative terms?

What is clear is that it is impossible to tell anything about the quality of a script, (that has not been 'datafied' in order to facilitate quantitative analytics) without reading and re-reading it, either as a whole or in part, since script quality is a value judgment based on the specific reader's interpretation of the script content they have read from their subjective perspective.

But if certain qualitative characteristics can be defined and a script "processed" to look for matches (or not) or correlations with these qualitative characteristics, then it should be possible to make meaningful, "first cut" qualitative judgments about a script before needing either to read it first or to read subsequent revisions made based on the qualitative characteristics revealed. At the very least this new process step would allow the script analyzer to "see" some of the "unread" content and "latent" potential of the script before even reading the actual text.

This is not to say that it is not necessary to read the script, far from it, only that much could be learned about the characters, narrative structure and genre conventions of the script, for example, before reading it so that when it is read, it may be read from a different start-point - one that reflects a level of evidential analysis that has already been done in advance based on the data.

Alternatively the script could be read first and then analyzed/visualized retrospectively using a tool so that the latter step could be used to more or less confirm or deny aspects of the "read" analysis or prompt the reader/analyst to go back and review certain aspects of the script again based on what the tool reveals. In this sense the analysis and visualization step can function both as a "pre-read" and a "post-read" diagnostic aid.

Idea Development

The screenplay business process can also be viewed as an idea development process or workflow whereby an original idea (whether commissioned or speculative) is developed from a raw idea into a series of 'deliverables' in the forms of a treatment, synopsis, rough draft, final draft, shooting script and finally transcript. In this scenario, the 'business' of the process is that of the accretion of content around the original idea kernel together with some kind of content transformation step at each of the process stage gates.

Screenwriting/Screenplay Activities	Innovation/Idea Stage Gates
Story planning (treatment, synopsis)	Envisioning idea
Story drafting	Expressing, Defining idea
Story rewriting	Enhancing, Enriching idea
First draft	Approved/Viable idea
Optioned Script	Evaluated/Valuable idea
Script polishing	Developed/Prototyped idea
Final/Shooting	Realized idea ready for manufacture/release
Transcript of movie	Product/service in use/practice

This idea management perspective emphasizes the importance of the first

stage of a screenplay lifecycle, namely ideation: the development of a script idea – the genesis or gestation stage. At this stage there are not necessarily any screenplay drafts or substantive script content of any kind. Just an idea of what the script is intended to be about. MacDonald (2004) has proposed the concept of a shared screen idea and a work group that engages with it, both of which are fundamental to the ideation, creation and realization stages of the screenplay lifecycle.

... I have argued that such practice congregates around a shared screen idea rather than focusing on a specific written text...I have also suggested the idea of a Screen Idea Work Group (SIWG) as a flexible and semi-formal work unit that congregates around the screen idea, and whose members contribute to its development.

Director Peter Jackson also refers to a 3-stage screenwriting process that resembles ideation and emphasizes the screenplay beginning as an imaginative work: (<http://gointothestory.blcklst.com/2011/04/peter-jackson-on-screenwriting-process.html>, accessed 04 September 2012).

We always find there are three distinct phases in the life of a film script. First, it exists before the film starts shooting. In this period, which can last from months to years, the script is a theoretical document—an imaginative version of the movie.

Then you start shooting and things come much more into focus—usually in a very positive way. We now have actors who bring their skill to the roles and suddenly we see the characters in a more vivid and tangible way...

The final writing phase comes in post-production, when the movie is edited. No matter what you were imagining when you wrote the script, and what you imagined during the shoot, nothing now matters beyond the actual cut film. We often find that script work continues during post, including writing and shooting new scenes, reorganising the order of scenes, or recording additional dialogue to slip into shots. We do all of these things, and the writing only stops when the film is finally finished.

Jackson's observations are important because they identify the dynamic nature of the screenplay, especially during production, when the screenplay naturally informs the filming but as Jackson makes clear, the opposite also applies. In this sense, both screenplay and movie are participating in a

valuable feedback loop. And managing this kind of feedback loop, once a screenplay is in production, is important functionality that many screenwriting applications simply ignore.

3.4 Screenplay as Social Network Anchor

Clearly the process view of screenwriting and the recognition of the collaborative involvement of many different types of stakeholders in that process confirm that a social networking perspective of screenwriting and the screenplay has validity. Here, the network anchor/context is the screenplay text itself and it is the changing text of the screenplay that, in effect, functions as the analog of the Facebook 'wall' or Twitter Tweet 'stream'. In this sense the screenplay text also functions also as a content hub to which other UGC can be attached to create social capital and further enrich the screenplay content and context.

A use case of a screenplay acting as the anchor for a social network and able to benefit from social networking activity, is in the educational domain in the activity of learning screenwriting, where each student is creating screenplays (or scenes within a screenplay) as a member of an undergraduate or postgraduate screenwriting class. Another example is the screenplay anchoring the movie production team or 'on-set' social network, where each practitioner tends to approach the script from their specific production-role point-of view (POV).

In both cases there are a range of social networking activities that we can expect members of these social networks to be engaged in, in these and other screenplay anchor use cases. One key activity is that of peer review for qualitative assessment of the content.

Clayton (2006) discusses the importance of good peer review practice in the design of her *MA in Screenwriting for TV and Film (Retreat Programme)*. This Royal Holloway (London) postgraduate course encourages the development of dyadic relationships, leverages collaborative 'group boxes' of four

students, and demands class peer review of student work, with some of this review work itself being tutor marked. Clayton points out that peer review can deliver 'insights for both donor and receiver' but she also highlights the risks of 'subjective' comment, the impact of 'negative' opinions and handling the 'conflicting' views of the peer group.

A screenplay is often the result of an act of collaborative writing (CW), which Henderson and De Silva (2006:3) describe as '*the process of multiple authors producing one document, by writing together and soliciting one another's opinions about their writing.*' The authors propose a narrative based business process model for CW, founded on Rhetorical Structure Theory (RST) so that the narrative can be divided within a team to help to enable the best people within a team to work on the narrative sections they are best suited to. They emphasize the importance of author rights and roles and more emphasis on narrative version control to facilitate the CW process. It should be noted that neither task allocation to roles nor version control capabilities are well supported in any current screenwriting application.

Takach (2006) approaches screenwriting from a knowledge management (KM) perspective proposing (section 4.1) that screenplays create a '*context for sharing*' and that the screenplay acts as a means for the screenwriter to communicate knowledge 'messages' to an audience so members can 'act' upon them. Sharing knowledge for learning is the primary objective of a 'community of practice' (cf. Etienne Wenger at <http://www.ewenger.com/theory/>), itself another way of describing the kind of social network that may exist around a screenplay, although it perhaps better described as community of 'interest' or 'purpose'. At different times during the screenplay development process, different communities of purpose emerge, comprising different participants and roles that have different priorities e.g. to sell the screenplay, to attract talent to the screenplay project or to produce a movie from the screenplay.

Summary

A screenplay that is either 'in-play' commercially or 'in-production' (i.e. principal photography has started) is at the centre of a social network, maybe even multiple overlapping networks, albeit rather smaller ones than a popular Facebook page or Twitter stream involve. As the screenplay progresses through its development and realization process, more and different stakeholders within the network make use of the screenplay for different purposes.

As the anchor of the network, the screenplay can become the focus for a number of social networking activities, like rating and commenting, and acts as a content hub that benefits from user generated content linked to the screenplay by the network participants. Today, most screenwriting applications do not reflect this social network dimension by providing the functionality required to allow a screenplay to be effectively co-created and socialized.

Datafication of screenplay content facilitates the perspective of screenplay as social network by making it easier to interact or engage with a script at a more granular level than the 'whole-script' document level. For example social network participants can rate and comment on, and attach enrichment content to, script content at the scene, role, location or even snippet level. This enables the generation of more social capital around the script which may lead to a qualitative improvement either in the screenplay itself or in the process of realizing the script as a movie.

4 Screenplay as Design

The prevailing design of screenplays is no longer fit for purpose.

Screenplay as design, as both a concept and practice is in need of a complete overhaul. Screenplay design is steeped in 20th century legacy practices of writing, publishing and distributing scripts on paper. The most obvious 'legacy' design element being the use of a Courier 12 fixed font typeface to simulate the production of a paper artefact on an obsolete device - the typewriter. The presentation of screenplay text generally assumes a printed rather than online textual artefact. Kindles and other mobile devices as a form factor and ebook-like formats as a delivery mode have largely passed screenplays by.

Even if a screenplay is only conceived as more like a web page where the text also acts as an anchor for linked content then the screenplay has the potential to become an interactive rather than static text. This interactivity can already allow individual scenes to be 'played' in different ways depending on the linked content (e.g. video and audio clips): for example for dialog to be spoken out loud or locations viewed via video clips rather than just described in scene headings.

This is not to deny the utility of the printed screenplay delivered on paper. Script supervisor Sabi Paisa offers proof of this in her blog post *Director's Homework* (see <http://scriptsuper.wordpress.com/2013/04/09/directors-homework/about> - accessed 17 June 2013) about the use of a printed screenplay by Director Ian Barry on the set of *House Husbands II*. As figure 4.1 shows, Barry uses a paper-based 'script book' to plan shots and generally annotate the script using his own personal colour-coded 'markup language.'

Personally I find this approach very compelling as a means to capture a single individual's ideas around a script using offline media (i.e. a printed book). Applications such as Peter Skarratt's Script Supervisor software (see <http://peterskarratt.com> - accessed 17 June 2013) can replicate some of this

script lining capability automatically but without the artistic flexibility that plain paper naturally offers (see figure 4.2).

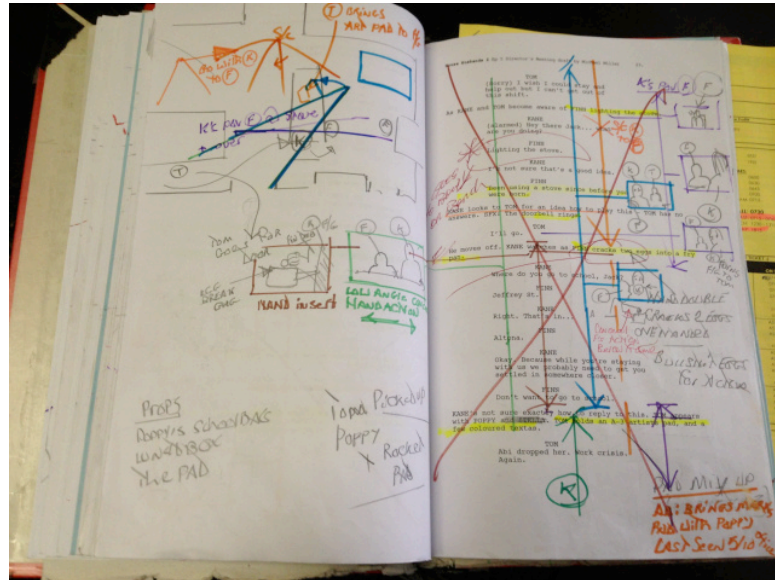


Figure 4.1 – Ian Barry's Script Book

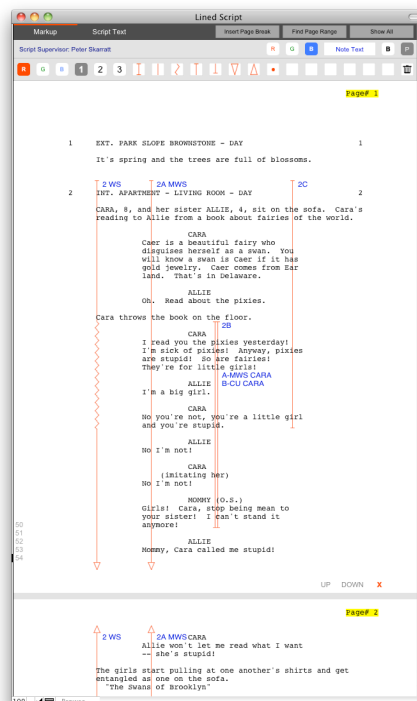


Figure 4.2 - A Lined Script in Script Supervisor

While a paper script is clearly portable and easy to annotate manually, an online script can offer the significant benefit of hyperlinking, like any web page. The interactive screenplay needs only the ability to add a hyperlink within the text or some other means to link content to screenplay data or metadata. These links may be to external web content (you can do this in any script written in any popular word processor (e.g. Microsoft *Word*) or to 'internal' content based on relationships inherent in the data structure of a datafied screenplay.

Examples of these external & internal links include:

- popup links e.g. to popup a character image from a character cue or a location image from a scene heading
- branch links e.g. to popup alternative scene choices at the start or end of a scene
- comment links e.g. to list a series of comments or Tweets posted to a scene
- storyboard links e.g. to view a series of board images linked to a scene
- clip links e.g. to play an audio/visual clip linked to a scene

This kind of screenplay linking creates a form of Augmented Reality (AR). AR layers 'virtual' information on top of a physical location. For example if you point your smartphone camera at a hotel on a street, AR may layer a booking form over the hotel. Your physical position, as determined by GPS, is what enables the layering on of content to augment the view you see. In a screenwriting 2.0 screenplay, the physical reality is the on screen text of the script, the virtual reality is the variety of content linked to the script

depending on the specific content location the reader is 'at' in the script. The GPS co-ordinates of latitude and longitude that drive the availability of AR content from the viewing perspective of a mobile phone camera become the scene and snippet, location or character metadata identifiers in a screenplay text. As a result, voice, image or video clip content could be layered over the text to create a more immersive screenplay viewing experience.

4.1 Data-Driven Screenplay Manufacturing

It is becoming increasingly clear that data can drive the design of almost anything. The emergence of computer-aided design (CAD) laid the groundwork for computer-aided manufacturing (CAM) whereby digitized design data is used to drive computer numerically controlled (CNC) tools. Automobile manufacturing is just one industrial process revolutionized by CAD/CAM and the use of industrial robots to carry out production tasks.

Now, the recent emergence of 3D printing shows that the potential of CAD continues to expand into the domain of direct digital manufacture (DDM). From gun parts to the bioprinting of living tissue (see <http://www.explainingthefuture.com/bioprinting.html> accessed 03 January 2013) 3D printing is widely expected to generate a new wave in CAD driven innovation at a personal rather than industrial level.

In 2012, I bought a Cubify 3D printer (<http://www.cubify.com>) and ran a series of four after-school class at Shaftesbury School (Dorset) in February 2013 to introduce a small group of students to 3D printing. We used Google Sketchup (<http://www.sketchup.com>) to design the objects we printed and Cubify's own software to convert the output from Sketchup into the input files required by the Cubify printer.

Anderson (2012) envisions 3D printing as enabling a new generation of individual craftspeople to participate in a new kind of creative activity – CAD/CAM for the person rather than the process. My brief exposure to 3D printing led me to consider whether the screenplay, and especially

screenplay metadata, cannot also be subject to some kind of ‘manufacturing’ process in the same way that coded CAD instructions can drive the printing of a 3D physical object. In the next two sections I discuss two applications of 3D printing techniques to the design and production of screenplays (not I emphasize, the output from them i.e. movies) that involves creating a slightly different kind of ‘prototype’ to those discussed in *Analysing the Screenplay* (2011) by Ganz (127-141) and Millard (142-157).

Subtractive Production

Subtractive production uses the digitized CAD file of an object to drive a process that reduces the volume of a source material in order to fashion it into the target object. For example, using a machine (e.g. a Roland Modela) to carve a curvaceous fruit bowl from a square block of wood. This has an application to screenwriting, specifically in terms of the process of adaptation.

As the term suggests, in the adaptation process the screenwriter takes a source text – such as a novel or theatre play text – and adapts it into a screenplay for realization as a movie. So how can subtractive production be applied here?

First we start with data, in this case, for example, the digitized text of the novel or play to be adapted. This data can simply be in the form of a plain text (.txt) file. Now when this data is fed into a specific type of content analytics tool and a specific technique (algorithm) applied, the tool can return a subset of data found in the text that is relevant to a screenplay adaptation process. In effect the tool takes the text (cf. the block of wood) and subtracts this screenplay-irrelevant data from the rest of the text (cf. the ‘scrap’ wood shaved off by the subtraction process) and leaves the screenplay relevant data (cf. a prototype fruit bowl object – not the finished object – I am not suggesting the output will be a complete screenplay).

The content analytics technique that applies here is not, for example,

predictive analytics or sentiment analytics (both discussed elsewhere in this thesis) but entity analytics. The focus of entity analytics is on identifying so-called 'entities' in the text: for example people, places and things as outlined by Seth Grimes in *12 Things the Semantic Web Should Know about Content Analytics* (2011: 3-4)

Entity extraction is a process that starts by finding entities in source materials, whether web pages, email, audio streams, images, or some other material of interest. Once discerned, the entity is disambiguated (Is "Ford" a car, an industrial company, an actor [which?], a theater, or a place to cross a river?). Then it is typed (Person, organization etc.), and (perhaps) mapped into a canonical form according to a controlled vocabulary. It may be designated with a uniform resource identifier that facilitates associating diverse information to the source material.

Now it just so happens that people, places and things have screenplay relevance since they map to (potential) characters, locations and props. So it's easy to see how running a text file through entity analytics creates a 'framework' for a screenplay by automating the population of, say, the character, location and prop tables in a screenwriting application that utilizes a database to manage datafied screenplay content.

Naturally you will not want all the characters and locations and props written automatically to the tables. But as entity analytics is also capable of counting the number of times these entities appear in the text, the adapting screenwriter can easily intervene and select say the top 20 or 50, or whatever, of each entity to write to the tables.

By using a subtractive production technique via entity analytics, a 'framework' of certain screenplay metadata is now available to use as the basis for the adaptation without the need to even read the novel or play that is being adapted. Furthermore it would also be relatively easy to link some or all of these identified entities to other 'rich content' located on the Web to kick-start the research process and link this additional content to the entities.

Additive Production

Unlike subtractive printing, 'additive' 3D printing builds 3D objects in layers: Very thin 'slices' of material are layered on top of each other from the bottom-up until the physical object is fully realized from the CAD plan that has been preprocessed into a stereolithography file (e.g. .STL file). Screenplays can be also considered from the same additive perspective (see figure 4.3).

For example, using a structural framework such as Vogler's Hero's Journey a screenplay can first be populated with a 'sequence' layer corresponding to Vogler's 12 sequences mapped across the three acts of Vogler's 'ordinary' and 'special' worlds. It is then up to the writer to 'add' the locations and scenes to 'flesh-out' these sequences. Similarly, Vogler's eight character archetypes can also be populated into the screenplay as 'template' roles and again the writer can choose to use all or some and name the individual roles as one or more characters, as appropriate for the story she is trying to tell.

Onto these layers is added the 'entity' layer comprising actual character roles and locations and individual scenes that involve these automatically added characters and sequences. The 'snippet' layer populates individual scenes with dialog and action description until the screenplay is fully realized and then more layers are added during production to create the script storyboard, shot-list or production breakdown reports for example. In this way a script can be viewed as being 'developed' subject to an additive or layering process.

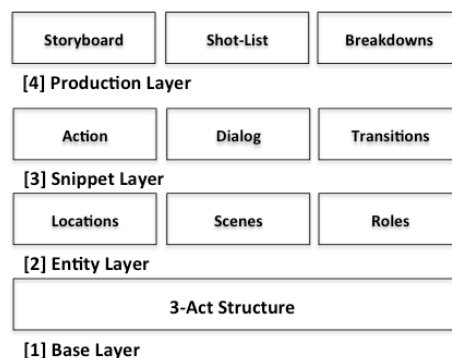


Figure 4.3 - Screenplay as Additive Layers

However, all this talk of ‘automated screenwriting’ is likely to be anathema to those for whom screenwriting is perceived primarily as a creative process and ‘manufacturing’ screenplays seen as a negative direction that will result in formulaic writing that delivers repetitive experiences for audiences, as Tom Gauld’s cartoon from *The Guardian* (review section 23/03/2013) in figure 4.4 suggests:



Figure 4.4 – Gauld’s Structural Journey

And many may regard this whole exercise as too trivial to care about. After all, the subtractive use of entity analysis hardly compares to the richly creative adaptation process presented in *Adaptation* (2002, Spike Jonze) and if the additive screenplay framework is based on a structure that has been characterized as a kind of ‘infantile bromide’ (Langford 2011:256) it’s hardly likely to be selected as a start point for a new screenplay. But the reality is that the subtractive and additive manufacturing techniques of 3D printing can be applied to screenplay metadata generation and perhaps over time, ‘auto-populating’ a screenplay with more or less metadata could become the normal way to start the ‘design’ of a screenplay in a Screenwriting 2.0 application.

Screenplay as Blueprint

Silver-Lasky (2004:53) characterizes screenplays as a 'set of instructions' others as a 'blueprint' (see discussion in Price 2010:44-47). Ganz thinks 'blueprint' is an inadequate metaphor (2011:1431) (but does see a role for the screenplay as a 'program for a sequence of interactions between the reader and the text' (2011: 131)) as does Millard (2010:14):

...it seems a less than ideal metaphor for the screenplay. The development of the screen idea inevitably involves collaboration, and therefore to concentrate solely on the screenplay as a source for the film-to-be seems unnecessarily restrictive.

Price (2010:45) is also not a fan of the blueprint metaphor saying, 'In its literal sense, a blueprint is a projection of a design for a material object' and that 'the insidious connotations of the literal meaning [of screenplay as blueprint] have proved persistently damaging.'

But how long can that continue to be a realistic attitude given the increasing sophistication of technology that can already leverage screenplay texts as design blueprint 'inputs' from which visual 'output products' are automatically generated?

Storyboarding and pre-visualization ('previz') software such as *Storyboard Quick* and *FrameForge 3D* already use screenplay data as the 'design' input for generating 2D storyboards or basic 3D animations from the textual screenplay content. The screenplay file is imported and a storyboard template or foundation pre-viz automatically produced from an interpretation of the screenplay data and metadata. For example FrameForge (see <http://www.frameforge3d.com/Products/Core-Version/> - accessed 09 July 2013) claim that: 'FrameForge Previz Studio can automatically generate blank sets for every unique location, and you can associate your stored shots with text in the script on a line-by-line basis...'

FrameForge then lets users additively enhance this pre-viz, for example by adding background images and 3D objects or people from its digital asset

library. Storyboards and previz animations perform a function analogous to a product prototype – lying somewhere between the product plan and the finished object (i.e. the filmed movie).

4.2 Screenplay Models

To date, the discourse around screenplay as design has been unduly focused on structure, which is only one element of design. For example, the sitemap structure of a website is integral to its design but perhaps more important to the audience (rather than the website's information architect) is the user interface or 'look and feel' of the site including the colours, icons and typefaces used.

The screenwriting structural debate has focused on popular 'paradigms' such as Field's 3 Act structure or Vogler's Hero's Journey and whether or not to proactively use acts and sequences and storylines as script building blocks. But, in practice, all these so-called screenwriting 'paradigms' are little more than ways of grouping scenes together into dramatic units, emphasizing specific plot points and character throughlines: they are not comprehensive designs but, as suggested above, outline 'metadata' frameworks, analogous to paint-by-numbers for artwork generation or the steel girders of a tower block construction to which everything else is connected.

There are other story models, at varying levels of granularity, which could be used to drive screenplay design. For example Soulier and Caussanel's (2002) model of the narrative and its narrative 'atoms': Situation, complication, resolution and result. Or Chris Huntley's *Dramatica*, a highly detailed theory of story (<http://dramatica.com/theory/articles/Dram-differences.htm> - accessed 25 February 2013). In Huntley's article *How and Why Dramatica is Different from Six Other Story Paradigms* (<http://dramatica.com/articles/how-and-why-dramatica-is-different-from-six-other-story-paradigms> - accessed 25 February 2013) he also outlines 'story paradigms' from Huage, McKee, Seger and Truby. All of which share

similar features including scenes grouped into sequences, sequences grouped into acts, character throughlines and specifically identified turning or plot points. Yorke (2013:256) has also provided a useful summary of 12 structural paradigms (not including his own version, the '3D Roadmap of Change' (2013:64)) as shown in figure 4.5 below:

	ONE	TWO	THREE	FOUR	FIVE					
TERENCE/FREYTAG	Set-up and call to action <i>Inciting incident</i>	Things go well Initial objectives achieved <i>Turning point</i>	Things start to go wrong as forces of antagonism gather strength <i>Midpoint</i>	Things go really badly wrong precipitating final confrontation with antagonist <i>Turning point/worst point</i>	Overcoming flaw Matters resolve for good or ill					
VLADIMIR PROPP	Villainy or lack	Departure	Struggle Victory <i>Midpoint</i> Liquidation	Return Pursuit Unrecognized Arrival	Difficult Task Marriage					
JOSEPH CAMPBELL	Innocent Call to world of adventure Childhood separation	Refusal of call Supernatural aid	Crossing threshold Atonement With father <i>Midpoint</i> Apotheosis	Refusal of the return Magic flight	Rescue Freedom to live Master of two worlds					
MAUREEN MURDOCK*	Separation from feminine Identification with masculine	Road of trials 2-headed dragon Slaying the ogre	Illusory boon Initiation and descent to goddess <i>Midpoint</i> Yearning to connect with feminine	Wild woman Healing	Integrating the feminine Beyond duality					
SYD FIELD	Set up	Confrontation <i>Pinch Point</i> <i>Pinch Point</i>		Climax and resolution						
VOGLER	Ordinary world Call to adventure	Reluctance or refusal of call Encouragement by mentor	Crossing 1st threshold Supreme ordeal Reward <i>Midpoint</i> Tests allies enemies	Pursuit on the road back 3rd threshold Death	Resurrection Return with elixir					
BLAKE SNYDER	Opening image Theme stated <i>Inciting incident</i>	Set-up Catalyst Debate Break into Act Two	B-Story Fun and games <i>Midpoint</i>	Bad guys close in All is lost Dark Night of the Soul	Break into last act Finale Final image					
JOHN TRUBY	Need/ Desire 1st reversal	Plan 1st Counter-attack	Drive Seeming defeats 2nd reversal	Audience revelation by ally 3rd reversal	Battle Moral decision New equilibrium					
FRANK DANIEL†	Status quo	The external want made explicit	Exploring the new world	1st big test overcome	Forces gathering	Hitting the wall	Desperate action	Success and aftermath		
LINDA ARONSON	Normality	Disturbance	Protagonist Plan	Surprise	Obstacle	Complications	Sub-stories	More complications and Obstacles	Climax	Resolution
CHRISTOPHER BOOKER	Call to action	Dream		Frustration	Nightmare	Matters resolved				
MICHAEL HAUGE	1. Set-up	2. New situation	3. Progress	4. Complications & higher stakes	5. Final push	6. Aftermath				

* Maureen Murdock, *The Heroine's Journey: Women's Quest for Wholeness* (1990).
† Frank Daniel was a screenwriting teacher who developed the 'eight-sequence' structure' (see Paul Joseph Gulino, *Screenwriting: The Sequence Approach* (2004)

Figure 4.5 – Yorke's Paradigm Summary

As Yorke's summary shows, the 3 Act structure has been adapted into 4 (3 act with a two-part second act), 5 (Shakespeare's favourite) and even 8 act structures, with lots of different interpretations of the role of each act in the narrative. Yorke also thinks that some of those who decry the influence of 3 act structures – he quotes Guillermo Del Toro, David Hare and Kaufman as examples (2013:xv) – 'protest to much' and 'however much they hate it...they can't help but follow a blueprint they profess to distrust'.

Another contribution to screenplay as design could be provided by the automated application of specific genre conventions to the screenwriting process to ensure that a screenplay 'feels like' other familiar, released screenplays considered to be in the same genre. Conformance to, or subversion of, genre conventions could be considered to be a design 'objective' that a screenwriter could espouse when writing a script. But then that also might lead to sterile writing if you agree with David Hare that 'All great work is now outside genre', as quoted in Yorke (2013: xv).

To discuss screenplay as design we must also consider two perspectives in particular: design patterns, as they apply to screenplays, and digital screenplay content manufacturing i.e. screenplay content generated automatically from a set of design patterns. The intention here is not to suggest that screenplay designs can entirely replace human creativity but that they may be able to automate some of the more mechanical aspects of screenwriting.

4.3 Screenplay Pattern Language

Christopher Alexander proposed design patterns in the domain of architecture in his book, *A Pattern Language: Towns, Buildings, Construction* (1977), in which he states that:

Each pattern describes a problem which occurs over and over again in our environment, and then describes the core of the solution to that problem, in such a way that you can use this solution a million times over, without ever doing it the same way twice.

(<http://www.patternlanguage.com/leveltwo/caframe.htm?/leveltwo/./bios/designpatterns.htm>, accessed January 15 2013).

The concept was subsequently adopted for use in software engineering in *Design Patterns: Elements of Reusable Object-Oriented Software* (1994) by the so-called 'gang-of-four' – Gamma, Helm, Johnson & Vlissides. An organization of design patterns specific to a particular domain is referred to as a pattern language.

A pattern can be described using a set of attributes, such as those outlined for software patterns by Maioriello (2002)

(<http://www.developer.com/design/article.php/1474561/What-Are-Design-Patterns-and-Do-I-Need-Them.htm> - accessed 03 January 2013).

Pattern Name	Describes the essence of the pattern in a short, but expressive, name
Intent	Describes what the pattern does
Also Known As	List any synonyms for the pattern
Motivation	Provides an example of a problem and how the pattern solves that problem
Applicability	Lists the situations where the pattern is applicable
Structure	Set of diagrams of the classes and objects that depict the pattern
Participants	Describes the classes and objects that participate in the design pattern and their responsibilities
Collaborations	Describes how the participants collaborate to carry out their responsibilities
Consequences	Describes the forces that exist with the pattern and the benefits, trade-offs, and the variable that is isolated by

	the pattern
--	-------------

Table 4.1 – Pattern Attributes

To see how this could be applied to screenplays, let's take the 3 Act structure and attempt to describe it in terms of pattern attributes:

Pattern Name	3 Act Structure (Field, 1979)
Intent	A means of dividing up screenplay content into a balanced set of scenes that reflect Aristotle's notion of a beginning, middle and end.
Also Known As	The 'paradigm'
Motivation	To enable an audience to understand the story world they are in, to follow the complications and conflicts of the story narrative and its characters, and to be satisfied by a resolution to the story narrative.
Applicability	Conventional linear narrative
Structure	Groups of scenes to reflect a one-quarter, two quarter, one quarter type content balance in the 3 acts with an 'inciting incident' scene somewhere around the middle of the first act and a 'mid point' scene

	<p>somewhere around the middle of the second act and a 'climax' scene somewhere around the middle of the third act.</p>
Participants	<p>All the main entities of a screenplay, namely: scenes, locations and characters.</p>
Collaborations	<p>The interaction of characters in both the action and dialog elements of the screenplay scene content.</p>
Consequences	<p>Complication of action and conflict of characters. The benefit of a 'balanced' script from a timing perspective for the audience. A trade off of structural predictability vs. structural novelty. Field's three key scenes as the variables isolated by the pattern.</p>

Table 4.2 – 3-Act Structure Pattern

On a similar basis, how about the notion of a 'protagonist' pattern?

Pattern Name	Protagonist
Intent	<p>A create a focal character in the screenplay whom audiences can relate to or root for.</p>

Also Known As	The 'Hero'
Motivation	To facilitate the ability of an audience to focus on following the narrative path of a single character through the movie from beginning to end.
Applicability	All character-driven screenplays that do not feature an ensemble cast.
Structure	A thruline that reflects the importance of the character to the narrative of the story.
Participants	Normally a single character or two characters acting as a team who in effect function as a single protagonist.
Collaborations	The interaction of the protagonist with other characters in the screenplay, especially the Antagonist.
Consequences	Complication of narrative events and character's composition and conflict of protagonist vs. antagonist. The benefit of a focal

	<p>character for the audience. A trade off of expectation of success vs. expectation of failure. Character veracity as the variable isolated by the pattern.</p>
--	--

Table 4.3 – Protagonist Pattern

Unlike many other design activities – manufacturing an aircraft or building a house for example – screenwriting does not typically involve forward-engineering or using a design pattern to drive the manufacturing of something. But there are some candidates for design patterns that could make up the pattern language of a screenplay to drive the manufacture of a screenplay in some automated fashion.

A key pattern is that of the scene and the scene groupings and characters that are instantiated to deliver structural paradigms such as Field’s 3-Acts or Vogler’s Hero’s Journey. These paradigms can be reused, as patterns, to manufacture the basis of a screenplay in Alexander’s words - *without ever doing it the same way twice* – where ‘doing it’ means populating the screenplay metadata framework generated from the pattern with scene content data - thus preserving the essential creativity of screenwriting.

If a writer sets out to write a ‘Hero’s Journey’ screenplay, he knows that potential elements of this pattern include a set of 12 sequences of scenes or ‘stages’ in Vogler’s terms (Vogler, 1996:14) and a prospective cast of 8 character archetypes (1996:31) plus the ‘hero’ (protagonist) and antagonist characters. So if I pick this pattern as a ‘design’ for my screenplay, I should at least expect that screenwriting software applications provide the means to use a template to generate an empty screenplay framework already provisioned with 12 sequences and 10 identifiable character types. These sequences in turn must then be provisioned with scenes by the screenwriter in order to enable the characters to interact. Like the concrete foundation and 4 corner posts of a building, this screenplay framework, generated from

a pattern, just provides a base to work on, the actual 'content' of the building could end up functioning as pretty much anything from an art gallery to a power station.

To my knowledge, no screenwriting software does this simple job of generating a screenplay framework based on a known pattern like the Hero's Journey (other than perhaps Chris Huntley's *Dramatica* – designed to specifically support his story pattern). Presumably the argument is that nobody actually writes screenplays this way or possibly many screenwriters agree with Langford's rather extreme opinion of the three-act structure as 'tyranny' and Vogler's Hero's Journey as 'infantile bromide' in *Analysing the Screenplay* (2011:256).

Baboulene argues, correctly in my opinion, that 'Structure is a consequence of our words, not a template for our words' and that 'Structure is an outcome of the creative process.' (2010: 32). In his opinion, structure is not useful to write a screenplay – as a design pattern - but as an analytic tool to help rewrite it. As he says (2010: 33): 'This is the main value of structure: fault finding and optimization after the creative outpouring is complete and we are in rewrite mode.'

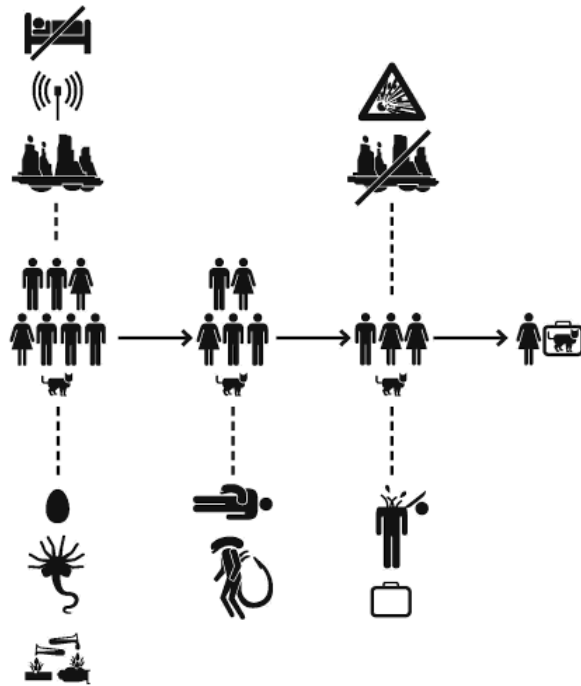
Alien Screenwriter Dan O'Bannon also offers the perspective of structure as an 'abstraction' from the text (2013: 22-23). He considers story structure to be 'an **invisible** construct that defines the relationships between parts of the story' and that 'The only way to detect a story's structure is for a knowledgeable person to examine the story and infer the structure from the story's visible parts. ' And because structure helps to show the screenwriter what's missing and where, O'Bannon regards it positively - as an 'empowering' tool for the screenwriter.

Yorke quotes McGovern's similar viewpoint (2013: 77-78): 'You write a script twice. The first time you pour out all your passion, anger, energy and frustration. Then you go back and write it with your head. '

Writing, or rather rewriting ‘with your head’, is where structure comes into play as a remedial rather initiating process. Cron (2102:76-77) is also not a big fan of story structures when used to mandate that ‘certain external events must happen at certain specific points in the story’. She considers this to cause stories to be written from the ‘outside-in’ because they cause writers to ‘craft **plots** in which these events occur rather than crafting **protagonists** whose internal progress depends on said events occurring.’

Baboulene, O’ Bannon, McGovern and Cron’s perspectives imply that these kinds of structural paradigms or design patterns are most useful to ‘reverse-engineer’ out of screenplays for analysis and rewriting purposes rather than ‘forward-engineer’ into screenplays for manufacturing or initial writing purposes. You do not ‘write to’ a structure but ‘rewrite into’ a structure. However, this argument puts screenwriting out of sync with most other kinds of manufacturing as few people in the real commercial world – and writing movies occurs in a real, commercial world - design output products or services with no specifically structured output product or service in mind.

But what these perspectives on screenplay structure do emphasize is that software could be useful as a means to facilitate the uncovering of structure implicit in the design. This surfacing of structure from the text, via diagnostic tools provided by screenwriting software, could provide a useful rewriting tool for the writer and a quick way for script readers and other ‘evaluators’ to easily ‘get’ the structure of a script. Perhaps this kind of analysis could generate a simple infographic visualization like this one that attempts to summarize the story of *Alien* using just a handful of icons:



[Allen] 9

Figure 4.6 – Alien as visualized by Milesi & Matteo (2012:9)

But it is probably unlikely to expect that screenwriting software will be able to automatically generate the rich infographic in figure 4.7 that charts selected episodes of the Inspector Montalbano TV series from 1994-2012 (see <http://cargocollective.com/federicafragapane/LA-LETTURA-CORRIERE-DELLA-SERA> - accessed 09 July 2013) anytime soon:

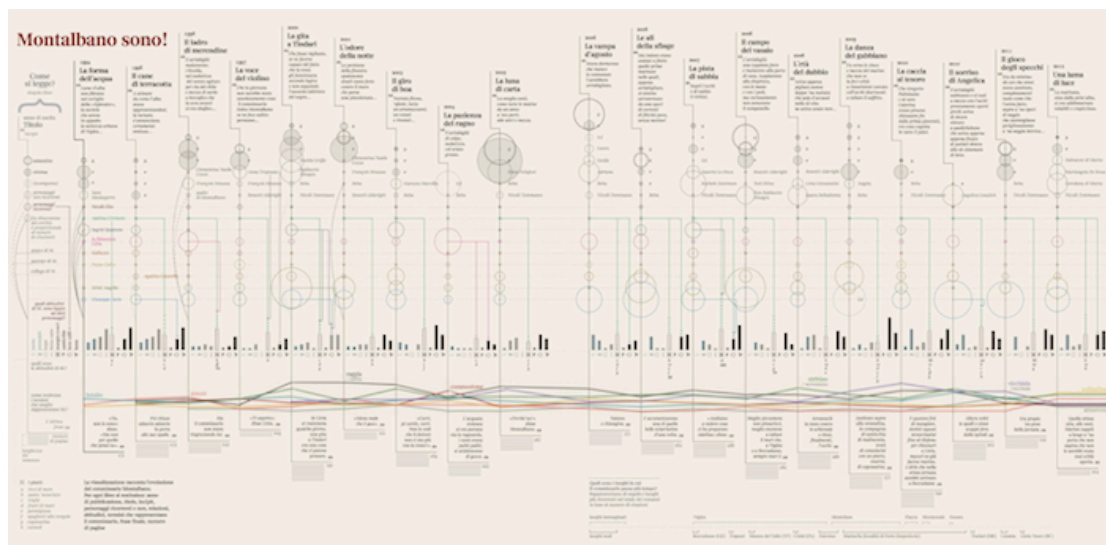


Figure 4.7 - Inspector Montalbano Infographic

Another key pattern, already ‘encapsulated’ in the pattern discussed above, is that of the Protagonist and Antagonist. Most screenplays have one of each and since this is a likely pattern, again it should be easy for software to prepopulate the script with a protagonist and antagonist characters. By doing this, the software helps to encourage the screenwriter to focus early on who these characters are, their motivation and backstory and the crucial interactions of these two principal character types.

Some screenwriting packages, like *StoryO* from Jungle Software Software, use a series of questions to help screenwriters to outline and flesh-out their story, its events and characters. The questioning paradigm is used to design the framework of the screenplay and ensure that the work is structurally ‘thought-through’ before any actual creative writing takes place.

So what we can already foresee happening is that by thinking in terms of design patterns as a means to manufacture screenplays, we can propose how the screenwriting software might help to automate and execute the implementation of these patterns in the application.

4.4 Reverse-Engineering Screenplay Design

Reverse-engineering screenwriting strategies and tactics from the released movie is a popular form of academic and movie critic analysis. But it’s important to remember that many versions of screenplays available for movies that are found on the Internet are not in fact the actual version of the screenplay that is reflected in the released movie. This version is only available when a transcript of the movie has been produced, since even a ‘shooting’ or ‘continuity’ script document is unlikely to reflect the final result of the film editing process.

Recently, tools have been released that deliver analytics directly from digital video. For example, digital analysis of storylines, say within multi-episode TV series, has been taken to the next step by the French researchers

Ercolessi, Sénac and Bredin and their web service StoryVisualizer or StoViz (<http://www.irit.fr/recherches/SAMOVA/ERCOLESSI/StoViz/#> - accessed January 15 2013). StoViz analyzes a digital video feed to identify and isolate storylines from the content by 'de-interlacing based on scene clustering'. In essence the software attempts to semantically group related scenes into storylines using a number of clustering algorithms. Therefore from a single feed of a movie's digital file, proposed storylines can be 'pulled out' and presented on their own timeline in order to better understand the various storylines encapsulated within a single video narrative.

This frame-by-frame video analysis approach is analogous to a Magnetic Resonance Imaging (MRI) scan, with the visual frame content equating to MRI scan's visual 'slice' content. Here a whole movie scan (cf. a 'whole body' scan), is used to discover storylines based on frame content in much the same way as a doctor looks for signs of disease or other anomalies in an MRI scan slice. The storylines are plotted onto a timeline in much the same way as the narrative of diseased tissue could be visualized from a series of slices from an MRI scan.

The screenshot from StoViz (figure 4.8) shows how the selected digital feed - episode 2 from US TV series *Malcolm in the Middle* - can be de-interlaced into 3 storylines, each of which can be played individually by the viewer user if required.

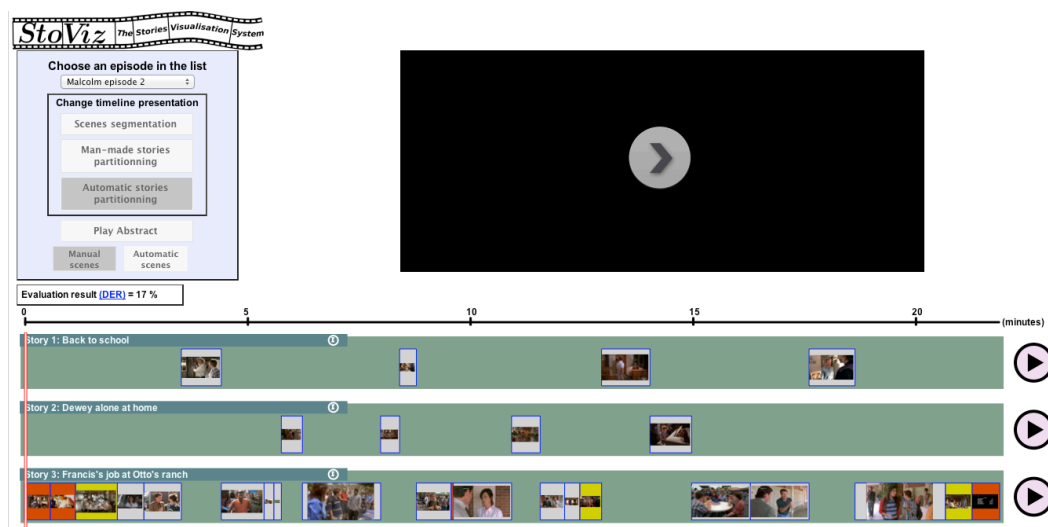


Figure 4.8 - StoViz Storyline Visualizer

Attempting the same reverse engineering of storylines from a screenplay text, as opposed to a digital movie file, isn't necessarily that much easier. The easiest way to identify storylines is obviously for the writer to flag scenes in the screenplay that belong to a specific pre-defined storyline and then provide a simple visualization of the scenes in that storyline and a means to 'drilldown' to the scene content at the click of a mouse. But what if the writer wants the software to identify storylines that 'emerge' from the text as opposed to those already defined within it?

Then, variants of some the techniques used by StoViz, such as speaker diarization and automatic speech recognition come into play. At a simplistic level this could mean that a potential storyline can be partially determined by commonality between scenes of:

- The characters in a scene
- The fact that characters speak to one another or take part in action together
- The scene's location
- The scene time of day and internal/external classification
- Props used in a scene
- The way characters speak (e.g. their vocabulary or syntax)

For example, the scenes that make up the storyline in *Alien* of the clash of wills between Ripley and Ash could be largely determined by the software identifying:

- Scenes that involve both Ripley and Ash with a preference for scenes that only involve these two characters
- Scenes that involve dialog snippets between Ripley and Ash
- Scenes that take place in Ash's 'domain' i.e. the sick bay

As with all analytics, even this kind of simplistic storyline analysis can be

used both to confirm that writer-intended storylines emerge from the screenplay text and to identify storylines that emerge from the text that the writer may have been unaware of and wish either to minimize or develop further in order to change the overall design of the screenplay.

Russell Chun's Story Visualizer is another tool that reverse engineers analytics from digital movie files. (see <http://www.russellchun.com/storystructure/storyvisualizer.html>, accessed 15 January 2013) According to Chun, what he does is:

“measure the level of excitement or drama over time by combining two indices. One index tracked audio levels, assuming that dramatic moments are accompanied by louder sounds (explosions, shouting, musical crescendos). The second index tracked changes in color, assuming that dramatic moments are also marked by rapid visual changes on screen (subject or camera motion, quick edits). The combined “drama” index is plotted to show the movie’s unique fingerprint”.

The screenshot in figure 4.9 shows a resulting plot from the analysis of the movie footage from a gladiatorial fight scene in *Gladiator* (2000).

In fact, this kind of analysis is more difficult to do from the screenplay text than from the digital movie file as many screenplays will simply not include enough identifiable metadata relating to audio levels or color changes to enable this kind of text analytics to succeed.

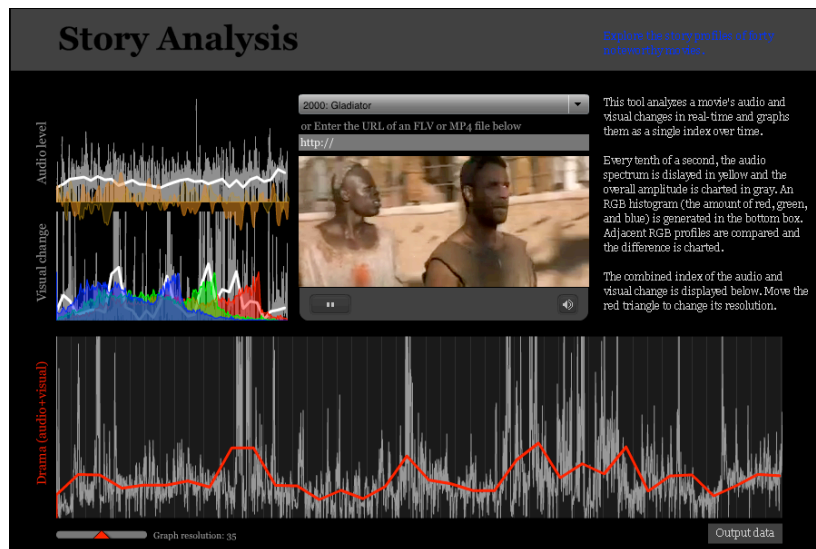


Figure 4.9 – Chun’s Story Visualizer

In fact reverse-engineering many of the de-facto standard strategies and tactics of screenwriting from a screenplay text is not that easy to do. For example, the ‘public’ demo script on my software deliverable Scenepad is an iteration of Dan O’Bannon’s screenplay for *Alien* (1979). *Alien* exemplifies many of the standard strategies and tactics of screenwriting such as:

- an inciting incident (the distress call that diverts their set course)
- a turning point (when the Alien emerges from Kane’s chest)
- raised stakes (the emergence of Ash as an antagonist to Ridley)
- conflict (Alien vs. humans, upstairs and downstairs crew)
- complications (Ash’s secret mission, Alien’s acid blood)
- climax (time-sensitive ‘escape’ of Ridley to the evacuation shuttle)
- resolution (Ridley safe in shuttle and Alien apparently destroyed)
- genre conformance (horror - monster loose in an enclosed space)

So how can these kinds of standard screenwriting tactics be reverse engineered from the screenplay text? As an example, let’s review *Alien*’s ‘turning point’ from this perspective.

4.5 Event Indicators

One structural analytic focus might be to try to locate a turning point in the screenplay by identifying what changes that is of significance – what linguistic marker can be defined as an ‘event indicator’ in the text?

In the past, manual textual analytics techniques, such as concordances, were used to analyze Shakespeare plays, today technology can help. For example, technology has made navigating texts by reference to a specific term easier, such as the mentions of ‘blood’ in Macbeth (see <http://www.shakespeare-navigators.com/macbeth/Blood.html>, accessed 4 March 2013). Statistics based on words used by Shakespeare are easy to find using sources such as Open Source Shakespeare.org (<http://www.opensourceshakespeare.org/stats/>, accessed 4 March 2013). Folger Shakespeare Library director Michael Witmore used DocuScope’s (<http://www.cmu.edu/hss/english/research/docuscope.html>) rhetorical analysis data mining technique to analyze vocabulary and syntax in Shakespeare’s First Folio corpus (see <http://www.fastcompany.com/1800987/data-minings-thing-shakespeare-takes-center-stage-digital-age>, accessed 4 March 2013). But these analyses do little to identify event notifiers.

A wide variety of sentiment analysis tools attempt to analyze text to determine its ‘polarity’ e.g. positive, negative or neutral or to find indicators of an emotional state e.g. ‘happy’ or ‘sad’. Apart from helping to define the general ‘demeanor’ of a character’s role, for example a positive or negative character or influence, or a happy or sad scene, if the sentiment analysis reveals a pattern of negativity or positivity after a specific scene then this itself may function as an event indicator, indicative of an inciting incident, turning point or twist or as indicators of a conflict ‘rollercoaster’ within the script text.

In linguistic studies, one kind of event indicator is a ‘discourse marker’ usually discussed in the context of discourse analysis (see Schiffrin, Deborah.

Discourse Markers. Cambridge University press, 1987).

Schilder's analysis of temporal discourse markers such as *after*, *before* or *while* (see <http://www.aclweb.org/anthology-new/W/W98/W98-0310.pdf>, accessed 29 March 2013) indicates that these marker words alone can be used as event indicators. According to Schilder, analysis of the clauses that follow these words may be classified to indicate a state, activity, accomplishment or achievement. The problem with applying this to screenplays e.g. *Alien* is that what follows the marker word in a script is usually not explicitly stated in an explicit syntactic unit e.g. 'After the alien popped out of his chest we all ran away' so any 'after effect' from the event is only implicit in the dialog or action text that follows. Or the marker word simply is not there to find, say to mark the event signifying the transition, for example, from pre to post Alien-loose-on-the-ship worlds.

Another kind of discourse marker may be the lexical cues to dialog acts discussed by Jurafsky, Shriberg, Fox and Curl (see <http://acl.ldc.upenn.edu/W/W98/W98-0319.pdf>, accessed 29 March 2013). Although *Alien* actually has a character who largely speaks in lexical cues (Brett) these kind of cues add little value as event markers in a screenplay. However they could be used to identify character alliances within a screenplay. For example, Brett's repeated use of the word 'Right' in connection with dialog involving Parker acts as a token representing agreement and is indicative of the fact that there is a (positive) relationship between these two characters (the below deck engineers) in the script. So isolating words like 'Right' and identifying the current speaker and the previous/next speaker could be a way to propose types of relationships between characters.

In this specific 'turning point' case the event marker that really identifies a change has taken place is that a new character is introduced – the Alien. And here, the release of the Alien onto the ship has profound significance for the rest of the script. We might be able to infer this from a character timeline because from this scene onwards a new pattern is introduced into the script,

namely that the characters that were active in the script prior to this new character introduction steadily become inactive (as they are killed off). So one event indicator of a turning point could be the introduction or loss of a character (or characters) that leads to a change in event patterns going forward.

A similar obvious event marker might be a significant change of primary location. Consider the famous cut to the Vietnam paddy-field in Michael Cimino's *The Deerhunter* (1978) switching us from the comfortable buddy-buddy life at home in the Pennsylvania steeltown to a landscape of death and destruction now surveyed by Michael's recently acquired thousand-yard stare. Vietnam has changed everything and the focus changes to action located in the 'special world' of Vietnam (or resulting from or influenced by it) rather the 'ordinary world' of the supermarket and bars inhabited by those left behind. Location change is often most obvious as a marker in the transition of the hero of the Hero's Journey from the ordinary world to a new unfamiliar world (and back) and the turning point that may be embodied in the 'call to adventure' or 'crossing the first threshold' may be identifiable simply from a major location change that follows it.

Other kinds of event indicators that could be used to identify a turning point or conflict and complications include a single unusual event (within the context of the screenplay as a whole) or a pattern of repetitive events – for example sex and/or violence. *Alien* also exemplifies a pattern of repetitive events after the turning point, namely the one-by-one killings of the crew by the Alien, that reflect complications and conflict that also lead to the crisis, i.e. Ripley is the only one left to fight off the Alien and prevent it getting back to earth. In Michael Winner's *Death Wish* (1974), a single and uncharacteristic (in terms of the script as a whole) sexual event – the rape – is the turning point that then sets off the series of violent, vigilante events that comprise the complications and conflict in the rest of the movie (a pattern that Winner had successfully used before in *Chato's Land* (1972)).

Summary

Screenplay design, as reflected in the primary debate around structural paradigms, is in its infancy. The design patterns of screenwriting are not yet well defined but some aspects of a screenplay's design can already be reverse-engineered from the finished movie by digital audio/video analysis tools. Although screenplays could be constructed both 'subtractively' and 'additively' it is unlikely they will ever be created in a similar fashion to a physical 3D printed object. In fact design may be something that it's best to infer from a screenplay rather than work to as a creative task.

Identifying aspects of screenplay design, through event notifiers in a text, could be a way to identify where important design elements such as turning points occur that may help a screenwriter to improve their use of these essential screenwriting elements.

Again, datafication of the screenplay content is helpful or necessary in order to facilitate the extrapolation of structure from content or to help identify event notifiers within the text.

5. The Road to Scenepad

For many decades the technology of screenwriting comprised a single tool: the typewriter. The courier twelve font of modern day screenplays is a legacy of the default fixed size font of pre-golfball typewriters; that all changed in the 1980's with the advent of the personal computer (PC). In the last decade, new form factors – the mobile phone and tablet – have become pervasive and stimulated the delivery of a number of new screenwriting applications adapted to suit the form and function of these new devices.

The era of pervasive personal and business computing is generally regarded as beginning with the commercial release of the IBM PC (or IBM 5150) in the USA in 1981. The PC's Microsoft Disk Operating System (MS-DOS) and later MS Windows were the operating system platforms that triggered the development of a rapidly expanding universe of personal and business software applications. And it was over the next decade that the first screenwriting applications appeared.

Write Brothers Inc. claims that it released the first 'screenplay formatting' program, called *Scriptor*, in 1982. However, one of the first applications to gain real market traction was *Final Draft* (FD) launched a decade later in 1992, which now claims to be 'the world's best and best-selling scriptwriting program. The industry standard for films, television shows and stage plays.' Whether this is true or not, FD is certainly a de-facto standard in that most other screenwriting applications can either import and/or export files in Final Draft's .fdx xml-based file format.

In fact there is no need to buy a specialist screenplay formatting program at all since most standard word processing applications can be used to write and format screenplays using the document template formatting functionality that comes as standard with leading applications such as Microsoft (MS) Word. Indeed the BBC's own *Script Smart* is a set of custom templates that 'add-in' to MS Word to facilitate script writing. These

templates make it easy to write, format and print/export a screenplay without the need to learn new software – but they do little else.

Early screenwriting applications are characterized by a focus on script formatting, feature-film movie writing and a single-user, writer centric paradigm. Over time more specialized screenwriting applications supplemented this generic functionality. For example *Movie Outline*, *StoryView* and *StoryO* that focus on planning screenplays and story outlining or *Character Writer* and *Persona* focused on developing an in-depth character profiles or a ‘character spine’ for a script’s cast of characters.

Some applications, like Chris Huntley’s *Dramatica Pro* and *Dramatica Story Expert* are specifically designed to support a single story development paradigm; in this case *Dramatica*, a substantive methodology for writing screenplays based on a four-act structure. So far no similar programs have been specifically developed purely to support a single popular story-writing paradigm, such as Christopher Vogler’s *Hero’s Journey* for example. In fact few screenwriting programs even contain specific built-in support for what is widely considered the most influential screenplay structural paradigm of all - Syd Field’s three-act structure.

As the screenwriting software market became more competitive from the end of the 1990’s onward, generic screenwriting packages offered variants or specialized functionality to service needs beyond purely writing, formatting and outputting the movie screenplay. In 2001 Final Draft launched Final Draft AV, claimed to be ‘the first software application specifically designed for writers of documentaries, reality TV, and commercials written in the two-column script format.’ Mariner *Montage* provided supplementary facilities for marketing your script (see McKie, 2006a) and Tim Sheehan’s *Sophocles* (now defunct) delivered a number of useful innovations for analyzing and visualizing script content such as charting and character relationship networks (see McKie, 2006b).

Scripts also became input for other non-writing applications designed for

movie storyboarding, pre-visualization or production management. For example, scripts can be directly imported into pre-viz and storyboarding software such as *FrameForge* and *Storyboard Quick/Artist* or into *Gorilla*, used for production budgeting, scheduling and storyboarding or *easySCOTT* for digital set management. The screenwriting application ecosystem grew to acknowledge the fact that the script has a lifecycle that extends beyond the writing or pre-production phase and into the production phase of movie-making. In 2012, John August released *Highland* a script conversion or file interchange tool that can import/export 3 key formats: Final Draft .fdx files, Adobe PDFs and the Fountain markdown format (also released by August's team).

The Internet, and now the widespread use of mobile platforms such as the Apple iPhone and iPad, also triggered a new wave of innovation in the screenwriting market. In September 2012 there were at least ten screenwriting applications available on the Apple iPhone/iPad App Store for writing or viewing screenplays. And new 'Open Source' and online screenwriting applications such as *Celtx*, *Trelby* and *Logline* are also proving popular, partly because a 'freemium' model ensures that they can be delivered for free initially, only requiring a paid subscription if users require more specialist features.

Celtx styles itself as a *media pre-production* application. It uses a project paradigm to manage the production of a wide variety of output products including film screenplays, theatre scripts and novels. Like its predecessor *Sophocles*, *Celtx* also includes production functionality to support scheduling and other production related reporting. Scripts can be stored online and projects shared with other users to foster collaborative working via the Internet. *Celtx* allows various digital assets to be attached to the screenplay to enrich the content.

I use the term 'screenwriting 2.0' to refer to application functionality and ways of working that reflect some different approaches to screenwriting than those generally available in contemporary screenwriting applications

or used in practice. In the design of my software deliverable Scenepad, I tried to incorporate functionality and practices that I considered to be missing or inadequately supported in most current screenwriting applications especially screenplay analytics and screenplay as social network.

It is the following brief survey of screenwriting technology that informs my experimental online screenwriting application prototypes, developed during my research, and my final application deliverable, submitted with this thesis, called *Scenepad*.

5.1 Screenplay Analytics Today

Screenplay analytics is the most obvious functionality missing from most of today's screenwriting applications, so this section reviews some analyses and visualizations already provided by the current generation of story-planning and screenwriting software. Note that these are generally not applications that store their screenplay content as data in a relational database (as Scenepad does) but store the content in some kind of file format as a document.

Format

The most common screenplay visualization is the presentation of a correctly formatted script for viewing on-screen. This is not an analysis of the script content as such, but is a visualization of the content according to prevailing industry conventions. Often this view is itself based on the writer selecting a specific format template to work with when they begin their new script. The script may be presented in an “uncluttered” full-screen view or as part of a workspace where it is surrounded by other panels that display content from within the script (e.g. a role list) or content linked to the script (e.g. images).

In figure 5.1 below, the script (on the right) is next to two navigational panels on the left. The upper-left panel offers a menu of functional options, and the lower-left panel displays a list of scenes in the script that provides

both a scenic breakdown of the script content and a navigation tool to allow the user to click on a scene heading and “jump-to” the scene selected in the body of the script on the right. This kind of multi panel presentation is commonly used in current screenwriting applications.

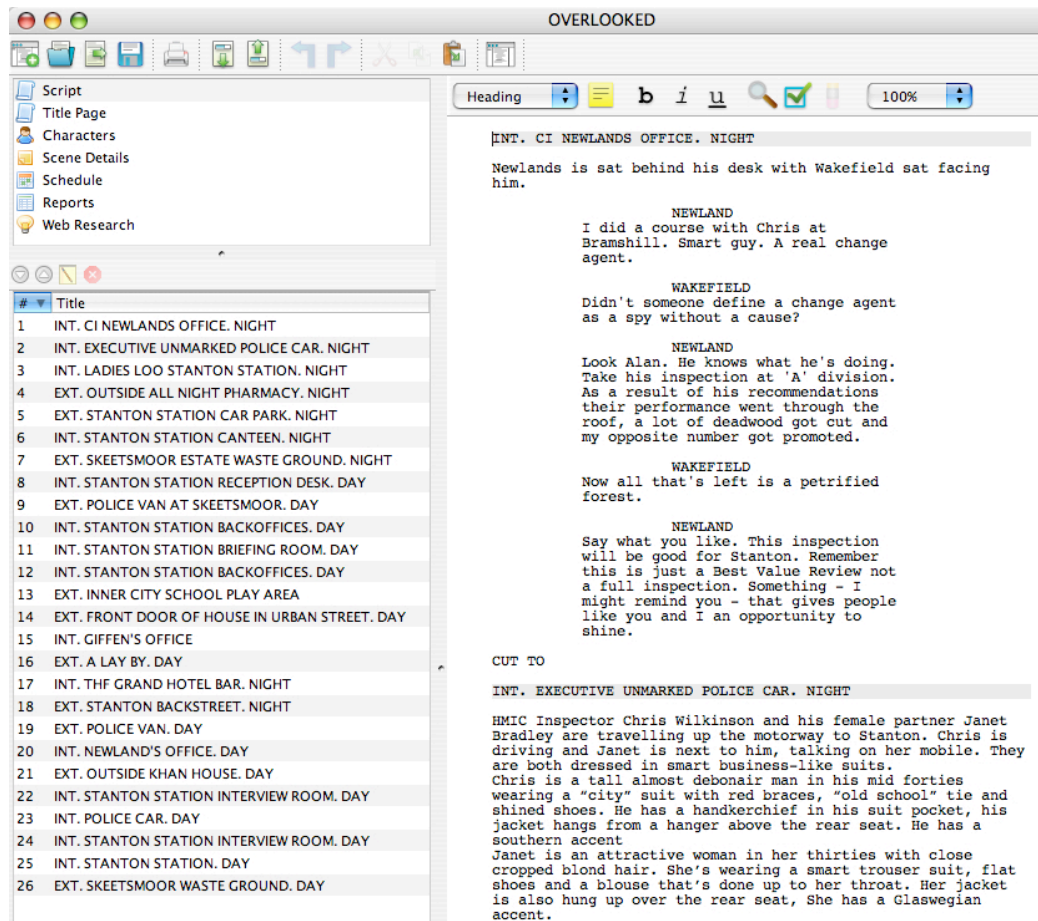


Figure 5.1 - Working with a Formatted Script in Celtx

Hierarchy

Another common visualization is of the script hierarchy (act and/or sequence and/or scene) in the form of an interactive, collapse-and-expand browser tree, which is used to navigate the script as a whole and to jump-to specific script content, such as a scene (see figure 5.2). By expanding each “node” or branch in the tree by double-clicking nodes with their mouse, the user can display acts, sequences and scenes. Usually when a scene is selected (clicked with the mouse) in the tree, the scene content is displayed in a

linked panel. These content navigation trees can also be used to browse the script content by entities other than scene headings, such as character names (the expansion listing scenes in which the specified character appears) or locations (the expansion listing scenes using that location).

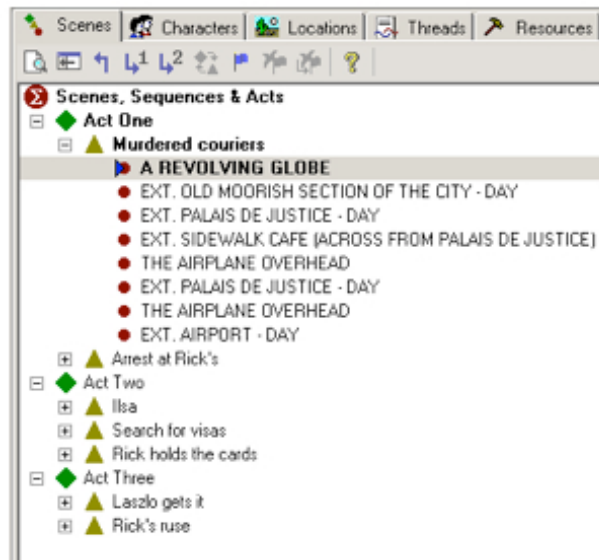


Figure 5.2 - Navigating a Script Hierarchy in Sophocles

Index Cards

Most leading screenwriting packages also provide index card visualization (see figure 5.3). Screenwriters have traditionally used physical index cards (e.g. in 3" x 5" format) or more recently post-it notes as a means to breakdown and visualize the screenplay narrative. Index cards are not just a way of visualizing a finished script draft in outline but also a way to plan screenplays, especially their structure and narrative flow, before any substantive scene writing begins.

For example, a card or post-it may represent a story-beat within a scene or a scene itself and may be color-coded to reflect the fact that it relates to a specific character-arc or plotline for example. Traditionally, cards are pinned to a corkboard or notes stuck on a wall/whiteboard so that the whole screenplay can be seen at once and individual cards/notes easily moved around to manually re-organize the flow of the script if required.

Screenwriting software can present script scenes in the same way on a virtual corkboard and make use of the mouse to provide interactive drag-and-drop card reorganization of the scene cards.

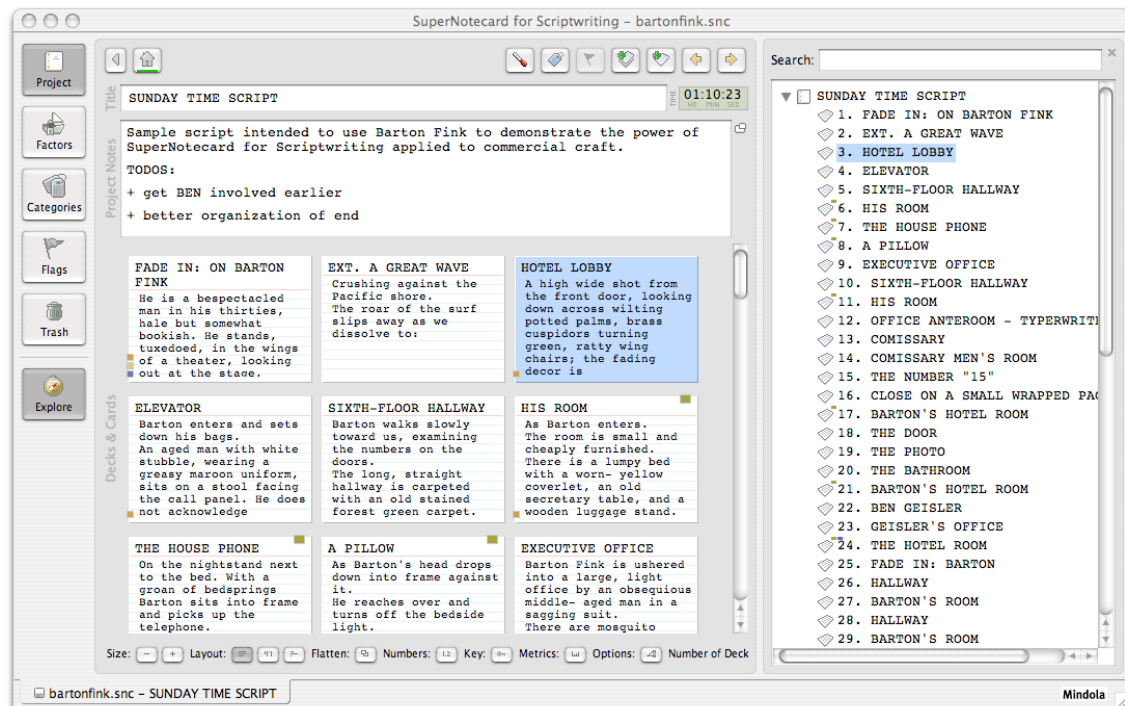


Figure 5.3 - Managing Index Cards in SuperNotecard

Charts

Some other kinds of screenplay visualizations use business-style charts to visualize script information. These include simple pie charts showing the proportion of DAY vs. NIGHT and INT vs. EXT scenes (see figure 5.4) or bar charts of scenes represented by number of words or the relative proportion of action vs. dialog used in a scene.

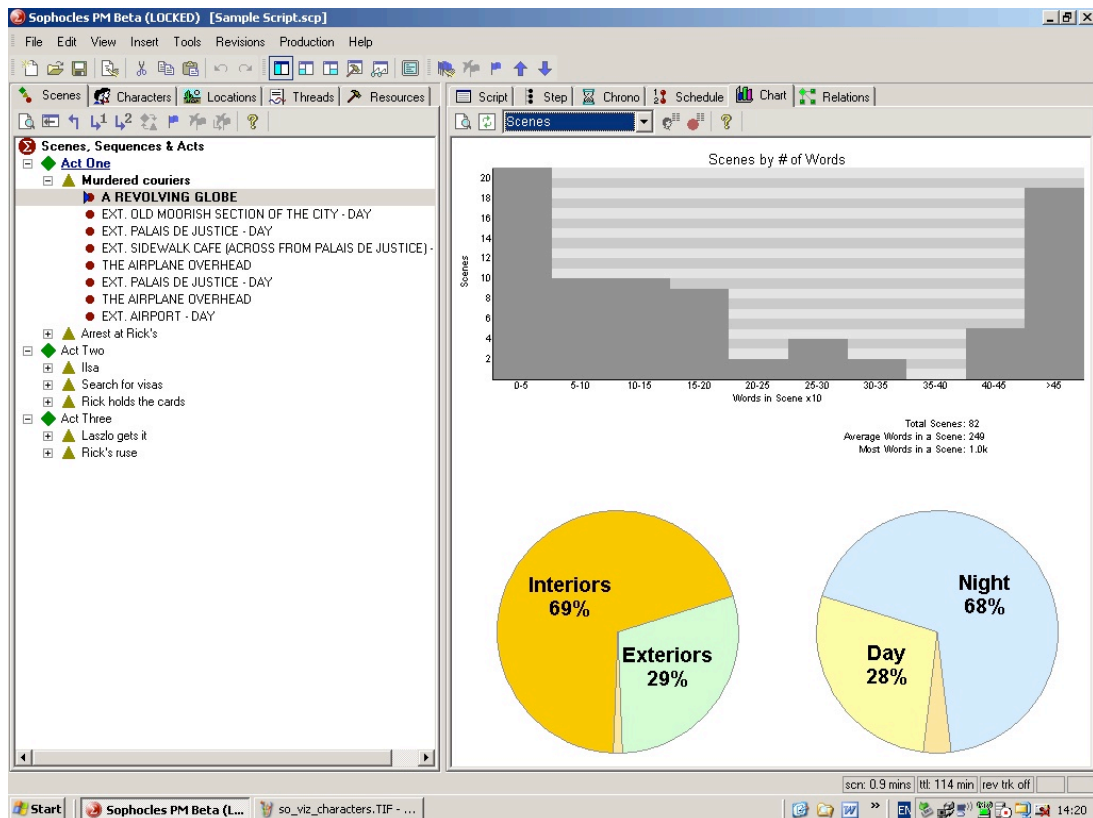


Figure 5.4 - Script Analysis Charts in Sophocles

On the basis of these charts, even without reading the script, one can tell that most of the movie *Casablanca* takes place inside and at night, which makes sense when you know that “Rick’s Place” is a nightclub. This charting may seem very simplistic, but even these visualizations may tell one something, or help one to frame questions, about a script, for example:

- More exterior night scenes could mean more production difficulty and expense vs. more interior and day scenes. Will this be an expensive movie to make?
- The lack of night scenes in say a horror movie might indicate that the script is deviating from genre conventions. Will this movie meet audience expectations?

Social Network

Another visualization provided by *Sophocles* is a form of social network showing character relationships and the size of each character node (represented in figure 5.5 as a circle) indicating the relative “size” of the speaking part of an individual character. Again, without reading the *Casablanca* script, one can tell that the key parts are those of Rick, Ilsa, Laszlo and Renault and that many relationship lines lead to Rick – so he is probably the protagonist (or antagonist).

If metadata was available to further identify the protagonist/antagonist and their “allies”, then this network visualization could be further improved by the use of color. Say **green** to represent the protagonist and allies and **red** to represent the antagonist and allies.

In this case the script-level social network is a useful overview of all the characters in the script but needs to be complimented by character-specific networks generated around a single selected character, that use the size of the “satellite” character circles to show how important the character is in relation to the “centre node” character. In the case of *Casablanca* this could be a useful way of visualizing Renault’s apparently dual loyalties: To his friend Rick and to his Nazi “boss” Strasser.

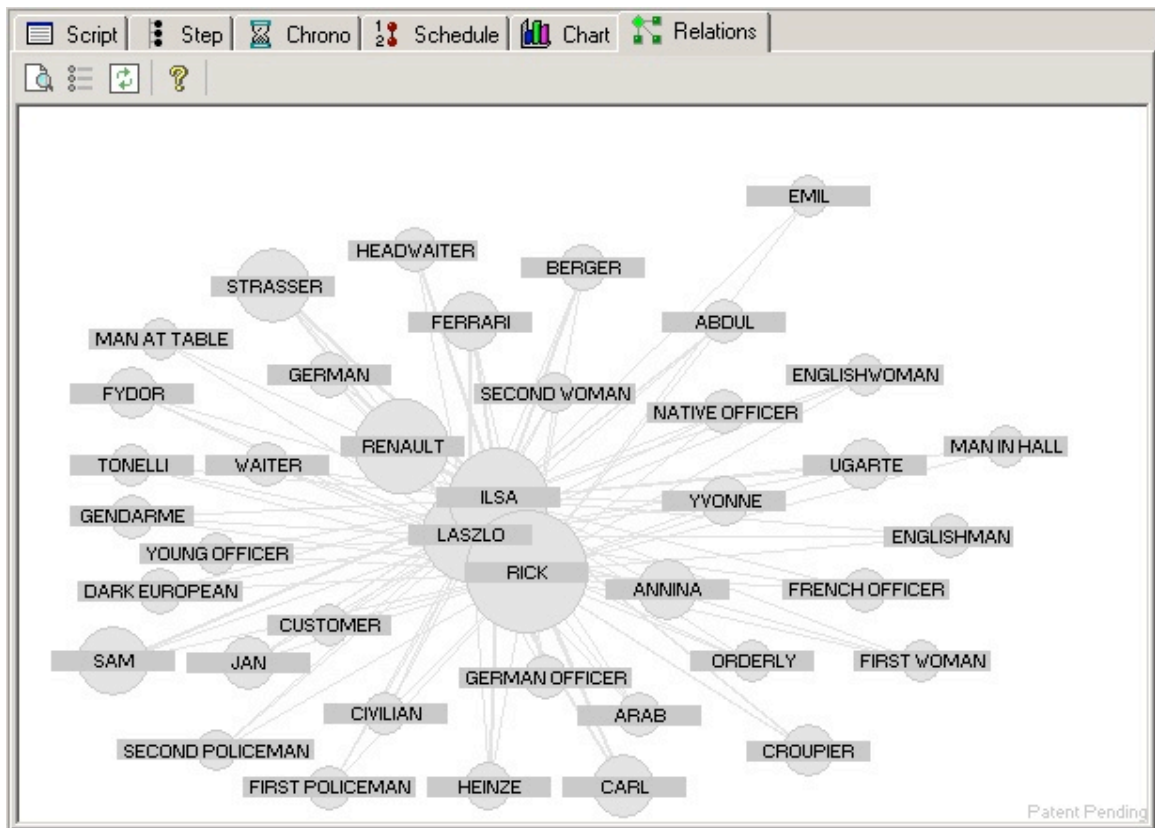


Figure 5.5 - The Social Network of Casablanca in Sophocles

Work by Agarwal, Balasubramanian, Zheng and Dash and by Gil, Kuenzel and Suen to parse screenplays in order to extract social networks from movie scripts provides much more sophisticated techniques than that used by my *ScriptFAQ* prototype that automatically generates a social network for a specific character in a script by using the vis.js library (see Appendix C).

Structure

Visualizations of structure are hard to find in the current generation of screenwriting software but Plotbuilder (www.plotbuilder.com) makes explicit use of what the application calls an ‘excitement graph’ as the basis for constructing your script by enabling the writer to place scenes onto specific plot points based on a graphic representation of a script structure (see figure 5.6) that clearly reflects Field’s three act structure ‘paradigm’.

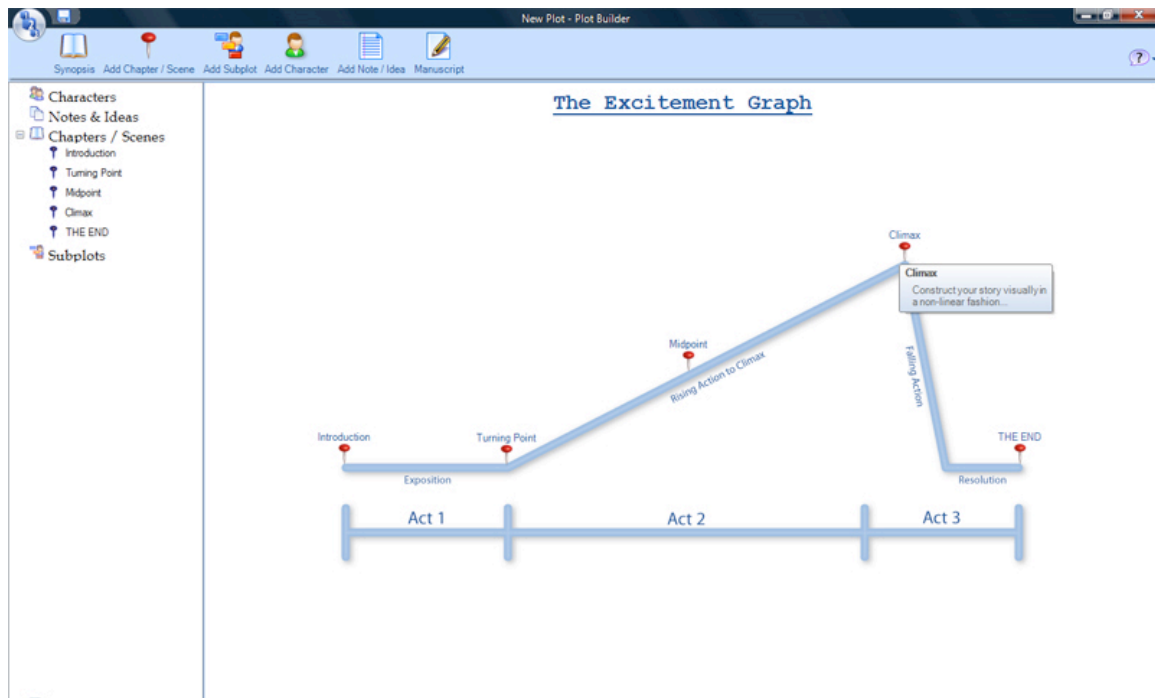


Figure 5.6 – Plotbuilder Excitement Graph

Timelines and Throughlines

Visualizations found in specialist story planning or outlining software such as *StoryView* (now known as *Outline 4D*), typically focus on the structure of a screenplay (e.g. the acts, sequences and scenes) against a movie timeline (above the structure in figure 5.7) and optionally include one or more “tracklines” (which would be below the structure in figure 5.7) used, for example, to display a character throughline or a plot throughline. This representation of acts, sequences and scenes as larger or smaller content blocks depending on their relative duration in the movie is subject to a US patent.

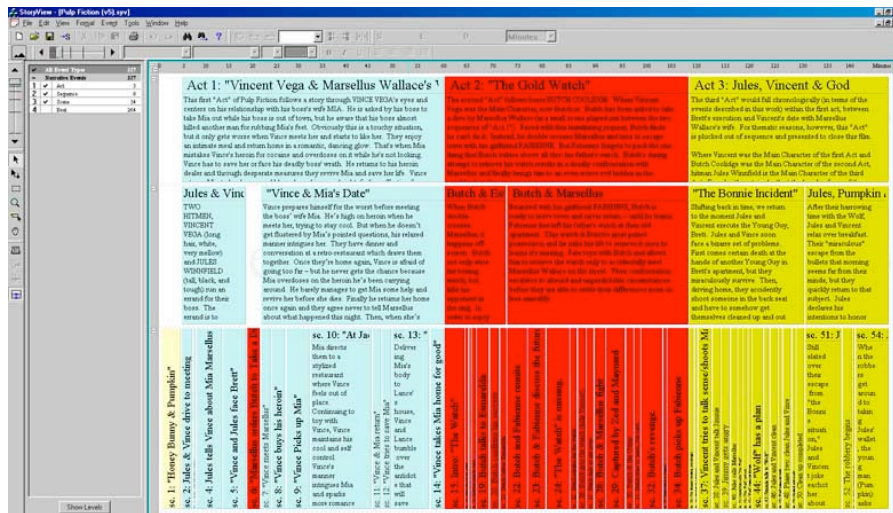


Figure 5.7 - The StoryView Structural View of Pulp Fiction

In fact StoryView allows the whole script to be printed in the form of a wall poster (see figure 2.26) to visualize the entire structure, complete with a timeline and selected/all tracklines.



Figure 5.8 - The StoryView Script Wall Poster

This combination of timeline, structure and throughline lends itself well to a particularly versatile implementation of screenplay analysis and visualization, as illustrated in figure 5.8. Here the visualization uses a framework that consists of structure (acts, sequences, scenes) and timeline “below the line” and selected throughlines “above the line”. So against a

constant below the line visual, users could select a specific type or set of throughlines to display above the line – for example plotlines or character arcs. So this one visualization-framing paradigm can generate a variety of useful timeline and structure based throughline visualizations from the script.

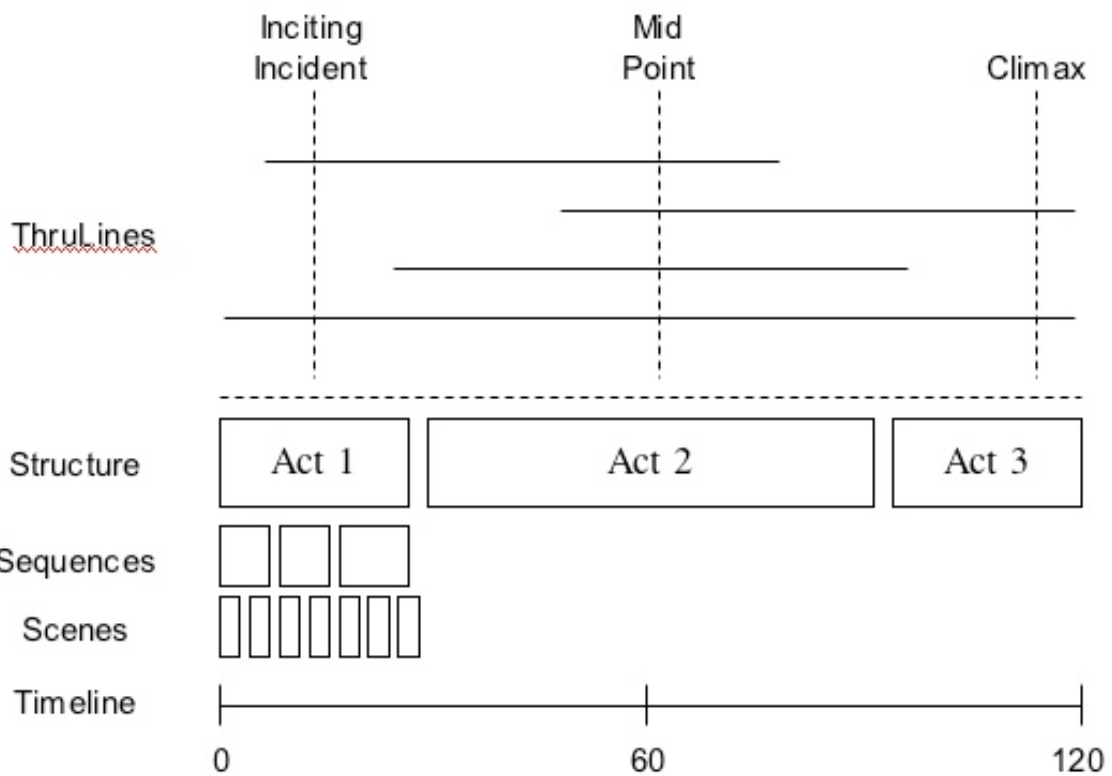


Figure 5.9 - A possible timeline, structure & throughline visualization

Visualizing the screenplay structure against an estimated movie timeline (e.g. based on 1 minute per page of script) helps the writer to understand if the script content is well balanced over the duration of the movie and whether act breaks occur at industry-acceptable points along the movie’s timeline to satisfy script reader and/or audience expectations. Visualizing the structure against a trackline helps the writer to understand if character or plot throughlines contain significant gaps that might somehow negatively impact the flow of the narrative or development of the character.

Storyboarding

Visualizing the script as a storyboard was discussed above as a form of pre-visualization of the movie that might result from the script. Usually this is done in a specialist storyboarding package but this function is provided in the Celtx screenwriting application (see figure 5.10). Note that that this storyboarding capability depends on the writer (or other authorized script collaborator such as a storyboard artist) manually adding the storyboard images to the script; the storyboard is not automatically generated in some way by the package itself.

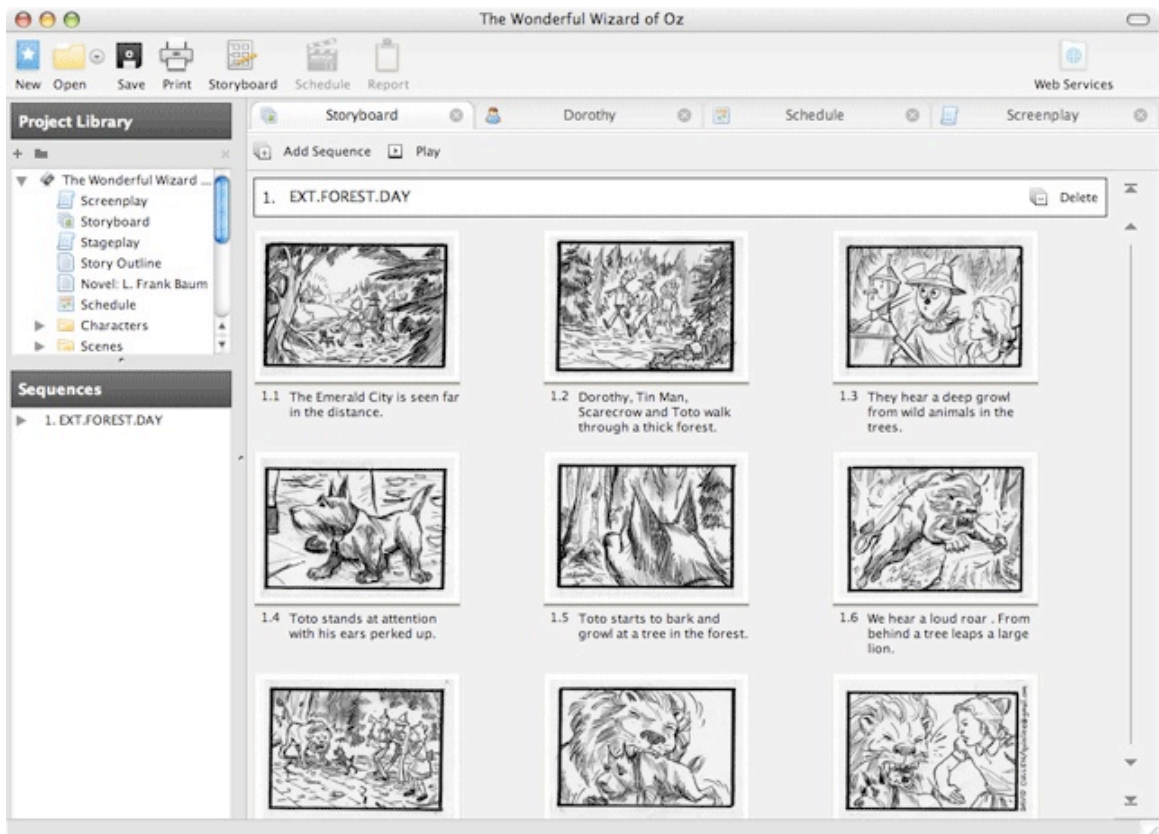


Figure 5.10 - Script Storyboard in Celtx

In 2013, Amazon (see <http://studios.amazon.com/storyteller> - accessed 11 June 2013) provided the ability for screenplays submitted to its Amazon Studios to have storyboards generated from and linked to the script content in a dual panel visualization comprising script text on the left and storyboard on the right (see figure 5.11).

As Jason Parham explains (see

<http://www.complex.com/tech/2013/06/amazon-storyteller-turns-scripts-into-storyboards> - accessed 09 July 2013):

So how does it work? Storyteller first scans a movie script uploaded to Amazon Studios, after which it identifies every scene, location and character, and then “casts them from a library of thousands of characters, props and backgrounds.” But if you're not totally satisfied with Storyteller's board, then you can manually recast, change locations and upload your own images.

What is perhaps of more interest here is that every script uploaded to Amazon Storyteller is being ‘datafied’ and that data refined by user intervention (i.e. to develop the storyboard from the initial auto-generated version). So Amazon is beginning the process of creating a ‘big data’ screenplay repository from which they may be able to derive interesting insights later on.

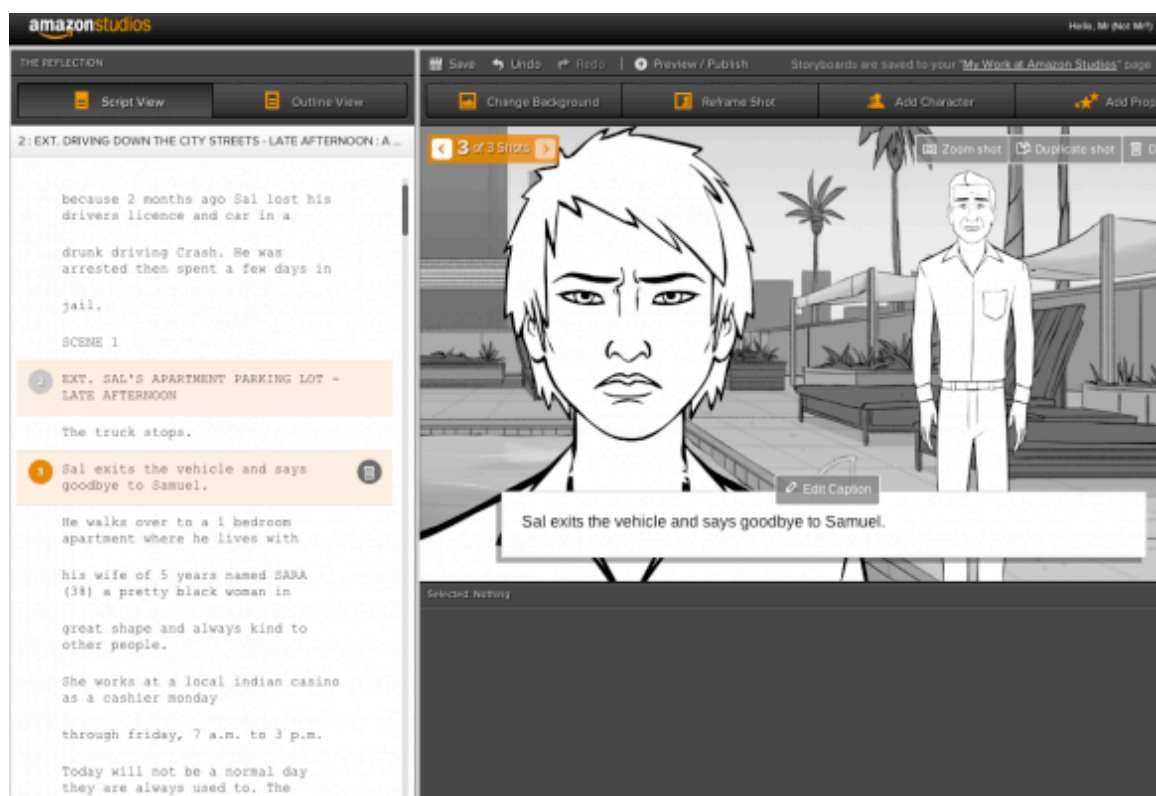


Figure 5.11 – Amazon Storyteller Storyboard

What's certain is that new generations of ‘screenwriting 2.0’ applications -

following the lead of past applications like *Sophocles* and my own *Scenepad* – are also likely to opt to datafy their screenplay content storage to provide more screenplay analytic capabilities.

5.2 Social Screenwriting Today

Other functionality generally omitted or poorly supported by traditional single-author, format-focused screenwriting technology is the social aspect of screenwriting, which manifests itself primarily in collaborative capabilities. For example, in *Final Draft*, there is a function called ‘Collabowriter’, which enables you to initiate an online session to view and edit a script and chat via instant messaging while you do this.

CollaboWriter allows you to collaborate on and discuss a script with other Final Draft users anywhere in the world via the Internet. One person initiates the session (the Host). The Host or another person can control the script (the Controller) while others view changes as they are made. CollaboWriter also contains a chat window so ideas and critiques can be shared instantly.

(*Final Draft* User Guide Version 8: 314)

This ‘session-based’ sharing was an early attempt by FD to provide some kind of collaborative script sharing that leveraged the Internet. But FD simply doesn’t have a concept of a ‘shared’ script that others can contribute to driven by role-based permissions.

Celtx has a concept of sharing a script ‘project’. You can link users to your project and give them permissions to access your project e.g. view only or change rights. Projects can also be ‘checked out’ – so that no changes can be made by other users while the project is in a checked out state. This is, in effect, applying basic document management principles to a screenplay script. The issue with this approach is one of granularity – it is the project as a whole that is being shared rather than specific entities within it or ways of working with it (see Borst, pp.223-225). But for many scriptwriters this may be ‘good enough’ sharing.

Most screenwriting tools do not recognize the need either for a script idea entity or a script work group to ideate it (cf. MacDonald, 2004). Script treatments, synopses and pitches may all be regarded as potential deliverables from the ideation stage of the screenplay development lifecycle (see figure 5.12) in that an actual script need not have been written in order to deliver any of them. To support ideation, screenwriting software needs the concept of a script idea, an ideation workgroup and a way of enriching ideas to support the ideation process.

And because many screenwriting tools are not multi user there is no support for the functional roles that are needed once you open up a screenplay to collaboration within a social network. If screenplays are defined as collaborative artefacts then a minimum of 2 roles are needed: View and Writer/Partner. The former only allows a user to view script content and add value to it (in the form of linked content) but not to mess with the core screenplay content (i.e. add/change/delete it). The latter may allow a user to add/change/delete specific or all script content. Obviously the additional availability of add/change/delete rights for the Writer/Partner role could permit more granular access to the screenplay content.

5.3 Screenwriting 2.0 Prototypes

The practical element of my research involved the creation of publicly accessible, online applications as prototypes for new screenwriting tools. I began my PhD as a non-programmer so initially a third party coded these sites, as acknowledged elsewhere. However in all cases the applications were designed, data-modeled in the MySQL database, tested and deployed by me. By the end of my PhD, I became an 'amateur' PHP programmer myself so my final prototype – Scenepad - contains a significant amount of my own coding albeit with some 'hard' coding (e.g. the script import function) done by someone else.

I briefly discuss and give examples of each prototype below. In each case I define the concept and aim of the prototype, the technology used and my

‘takeaways’ from each prototype used to inform my final deliverable, Scenepad.

In this section I also refer to two ‘benchmark’ commercial screenwriting applications: *Final Draft* and *Celtx*. These have been chosen for specific reasons. Final Draft is the de-facto market leading screenwriting application, and (as of 2013) remains a traditional software package in that it must be downloaded and installed on your local PC/MAC and costs significantly more than the free or downloadable ‘apps’ so prevalent today. Celtx is the most sophisticated of the newer open source screenwriting applications, offers a ‘freemium’ sales model – free to start, pay for premium features - and runs both offline and online. My references to these applications will be to specific versions namely Final Draft 8 and Celtx 2.9.1.

The practical deliverables discussed below are:

Year	Application	About
2006	Scriptcloud	An online application for generating a frequent words text cloud from the text of an uploaded text file of a screenplay.
2007	Scriptgeist	An online application for generating various text clouds, charts and lists from the text of an uploaded text file of a screenplay.
2008	Contentcloud	An upgrade of Scriptcloud that allows comparison of two text clouds from different texts.
2010	Sceneclass	A Wordpress plugin for managing a group-based script/scene writing class within a Wordpress

		blog.
2011	Scenewrite	An online screenwriting application for use in the pre-production and production phases of a script's lifecycle.
2012	Scenepad	An online screenwriting application for use in the pre-production and production phases of a script's lifecycle.
2013	Scenepad3	A variant of Scenepad with a 'responsive' UI that should work more effectively on tablet devices.
2014	ScriptFAQ	A script datafication program focused on delivering screenplay analytics based on a 'frequently asked questions' paradigm.

Table 5.1 – List of Prototypes

5.4 Scriptcloud [2006]

My first prototype was Scriptcloud. Scriptcloud launched in March 2006 on scriptcloud.com. Hannah King coded the application in PHP using my design and data model, managed in the MySQL database. As figure 5.12 shows, at retirement in July 2008 the site had 276 active users who had uploaded 343 scripts as the source for text clouds.

The idea behind scriptcloud was to allow screenwriters to create a word frequency cloud from the text of their screenplay, in a similar way to creating a 'tag' cloud from the tags linked to blog posts. The aim was to understand if this simple analysis and visualization had any analytical value in terms of helping the writer to understand their screenplay better.

The screenplay is not datafied, instead an uploaded text file is parsed and the scriptcloud generated from the words parsed from the text file. The database was used to store metadata about the script (e.g. title, genre) and the resulting cloud image (fore easier retrieval later) not to store the words in the script.

The site is currently located at: <http://scriptcloud.tripos.biz>

Scriptcloud requires users to register to upload a text file of an English-language screenplay into their personal script 'library'. On upload, a word cloud is generated that is stored in the database and can be downloaded to a PDF or simply copied to paste somewhere else. The user can then access their clouds from their library, on-demand. I added hundreds of popular movie scripts for reference purposes and made these accessible to all registered users.



Figure 5.12 – Scriptcloud Statistics July 6, 2008

Keyword Text Clouds

Scriptcloud leveraged a visualization paradigm that had recently become popular on the Internet: The tag cloud. A tag cloud is used to visualize the textual metadata ‘tags’ added to a piece of content such as a web page or a blog-posting for example. The purpose of the cloud is to give an indication of the popularity of a specific tag in terms of the body of content the tag has been used to refer to and to help categorize content for searching and navigation purposes. Tag clouds are often used to highlight changing patterns of “buzz” among large and dynamic data sets, such as active news blogs. Typically, tagged data sets could comprise daily blog postings, new web pages or images or videos added to a site such as Flickr or YouTube.



Figure 5.13 – Scriptcloud.com Home Page

Conventionally, a cloud displays more popular tags in a bigger font and/or in a specific colour. The cloud itself is usually limited to displaying a top-n number of terms (e.g. the top 50 tags from a specific content set) and tags representing a single language. Tag clouds can also be used as navigational

devices. If a tag word is “live” (i.e. represents a link to a web URL) the user can navigate to a web site or page or blog post or specific instance of a term within the content set, simply by clicking the tag.

Although a script as a whole could be tagged with specific metadata, such as the genre of the script for example, and a tag cloud produced from a set of scripts tagged in this way my intention with Scriptcloud was to show how cloud visualization can be used in a different way – potentially to surface themes on the script.

A ‘Scriptcloud’ uses the textual content of a script, rather than any user-added metadata tags, to produce what is more accurately termed a “keyword cloud” of the script content itself. The default scriptcloud generated from a user-uploaded script (see figure 5.14) shows the 64 most popular terms used in the script (after an exclusion or stop list is applied to filter out common “noise” words e.g. “a”, “the”, “and” etc.).

The user also has the capability to produce clouds from their script that exclude additional user-specified terms (e.g. profanities) or include only specific terms e.g. character names in the script. What a scriptcloud represents is a very limited, but potentially interesting, linguistic ‘fingerprint’ of a screenplay. Users have reported that it can help them to identify or recognize keywords or themes in their script. Their attention has also been drawn to overused nouns or verbs in the script, prompting them to be more inventive in their choice of vocabulary.

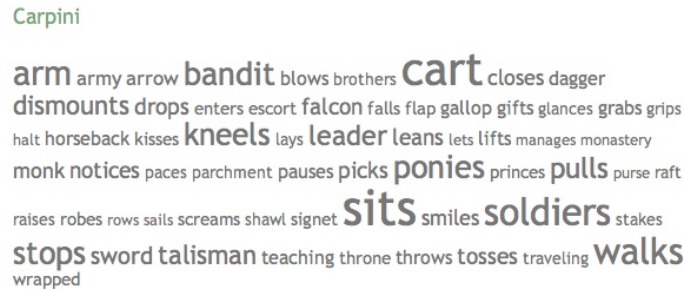


Figure 5.14 - Scriptcloud of My Own *Carpini* Script

Although the scriptcloud produced from individual scripts is interesting, what is more interesting is what clouds presented from the entire corpus of scripts could illuminate. For example, the range of keywords presented in a scriptcloud already excludes a large list of noise words. But it turns out that most scriptclouds show that the most popular keywords in a script are banal action terms such as “sits” or “walks” or colloquial dialog terms such as “gonna” and profanities such as “shit”. These terms are seldom keywords and often tell us little or nothing about the thematic content of the script.

Producing a scriptcloud from a corpus of scripts is likely to produce a cloud full of relatively banal terms that can then be iteratively added to the general exclusion list so that subsequent scriptclouds of the same corpus are likely to be that much more representative of genuine keywords in the corpus. Once the corpus includes a large enough sample of scripts within a specific genre then a genre-specific scriptcloud can be produced that also facilitates the creation and use of a genre-specific exclusion list. Then when a script is uploaded and linked to specific genre metadata, this genre-specific exclusion list could be used as the default stoplist rather than any generic exclusion list.

As noted above, each script uploaded to the site is also tagged with a genre classification. If a significant enough content set were established on a site like Scriptcloud, new kinds of scriptclouds can be produced that indicate the most popular genres and produce a fingerprint of the most popular terms across all scripts tagged to a specific genre. This could be useful commercial

information for a screenwriter: On the one hand to indicate current popular genres and on the other to enable the writer to compare their individual script's scriptcloud to the aggregated scriptcloud produced from many scripts sharing the same genre as their own script.

In practice, Scriptcloud generated minimal useful feedback from the hundreds of users who uploaded scripts to the site. Scriptcloud, like most sites that leverage user-generated content (e.g. Facebook and Twitter), depends on user engagement to succeed and deliver real value. The more content that is uploaded, i.e. scripts in this case, the more that can be done to leverage that content. The majority of users uploaded a script, got their scriptcloud and never came back.

In any case, the usefulness of a Scriptcloud as a script analysis tool is limited since it depends largely on the composition of the exclusion or stoplist. Too many words in the stoplist and keywords that might be indicative of theme could be lost, having been excluded from the cloud. Too few words in the stoplist and keywords might be crowded out by noise words. So some algorithm would be needed to optimize the stop list to reduce the impact of either scenario on the cloud generation.

Also it is likely to be coincidental that a scriptcloud does in fact provide some thematic clues to a script's content. For example, the scriptcloud generated from a version of the scripts of Coppola's *The Godfather* (1972) surfaces the word 'family', which many would consider an important thematic element of the script. However, it would be hard to tell from the surfacing of this keyword that *The Godfather* is not about the humdrum life of some regular 'mom and pop' family but the violent saga of a Mafia clan.

Godfather, The

anymore **arm** bastard bathroom blinds bodyguards booths checks **closes**
courtyard crosses dials drops embrace **enters** exit falls **family**
finishes gate glances gotta greets guarantee guarding **hangs** hesitates **hospital**
hotel hurries kisses knocking leans lifts limousine **love** luparas **nodes**
opens peeks picks pours **pulls** purse pushes **refuse** rows rushes scream
settles shakes **sits** smiles speeds **stops** strength throws tries
walks war wears weeping whispers wrapped

Figure 5.15 - *The Godfather* Scriptcloud

Refining the stoplist to exclude more of the most frequent words from the initial scriptcloud is a way to further isolate 'suspected' keywords. In the case of *The Godfather*, it also generates a useful result that is almost text-book 'apple-pie' in its Americanness: America, family and love. Yet arguably the 'genco', 'sicily' and 'war' terms give some clue that this script is not likely to be about some Norman Rockwell family.

Godfather - Some Keywords

america **family** genco **godfather**
love sicily war

Figure 5.16 - *The Godfather* Keywords

But what if a theme is not easily identifiable from an actual word in the script? For example, one might propose that one theme of Friedkin's *The French Connection* (1971) is 'persistence' - as in Popeye Doyle's persistence in getting his man, aka 'the frog'. However the term 'persistence' does not appear in the scriptcloud and neither does anything that clearly relates to this term. What does surface though are a number of active verbs that suggest this script might be some kind of action movie.

French Connection, The

arm arriving ashcan **ass** bathroom blows **booth** cab candy chase climbs cop
corridor crosses cup des **doorway** elevator **etc** exit falls fires garbage
glancing gonna headlights heroin hotel jam jumps junk kilos
leaning leaping moi nodding **pan** pants pizza plunging pour profile
pulls pursuit pushing raises removes ripping **roaring** scream screeching
shake **shit** slips **smiling** spinning **subway** surveillance **throws** toilet tow
tracks windshield zoom

Figure 5.17 - The French Connection Scriptcloud

When you've seen the movie, then the surfacing of words like 'heroin' and 'junk', 'chase' and 'pursuit' and 'subway' for example, are significant but not especially illuminating if you haven't seen the movie. We get no clue that the movie is set in New York city and is a battle between a couple of dogged NYPD vice squad cops and a French drug lord.

Scriptclouds are a simple, interesting visualization of the screenplay content that might just surface, through the words used, something interesting that confirms or denies a screenwriters intention. It would be easy enough to include a scriptcloud as part of any screenplay package that is provided to readers since it does provide some clues as to what the script might be about. On that basis there seemed no reason why they should not be included in Scenepad.

5.5 Contentcloud [2008]

To further develop the potential of scriptclouds, the next version of Scriptcloud was Contentcloud, a site that focused on another viable use for scriptclouds: for text comparison purposes. For example to compare one version of a script with another - either drafts of the same script or sequels - or one 'act' of a script with another within the same script. To facilitate this new usage, an update to Scriptcloud was produced to allow for side-by-side comparison of 2 sets of content. The scope of Scriptcloud was also expanded to allow for the upload of any type of text file, not just a screenplay. For example, this enhancement enabled the text of a screenplay to be compared to that of the novel it was adapted from.

This site was called Contentclouds.com and was again coded in PHP and MySQL by Hannah King based on my design and data model.

The site is currently located at: <http://contentcloud.tripos.biz>

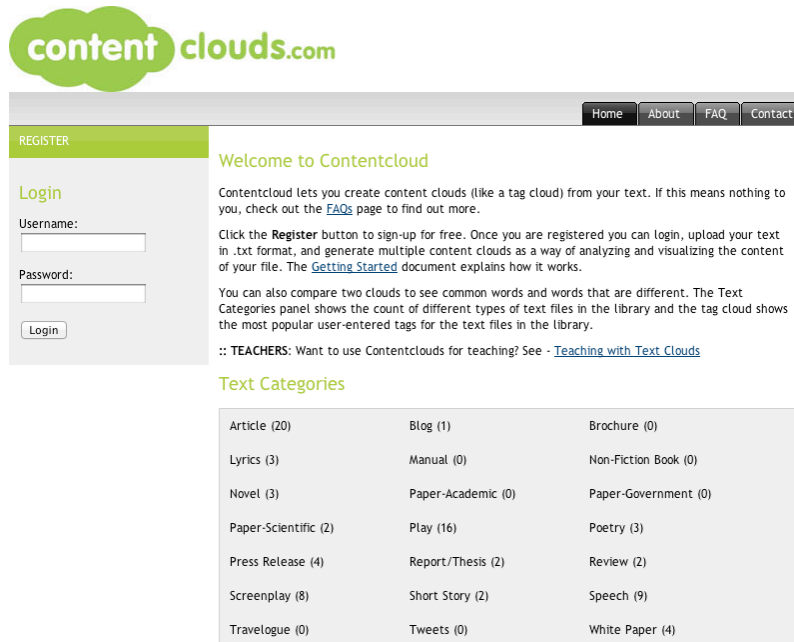


Figure 5.18 - Contentclouds Home Page

In the comparison of the screenplays of *The Godfather* (1972) and *The Godfather II* (1974) in figure 5.19, it is clear that the 'theme' of family still runs through the text. The Contentcloud comparison also includes words shared and not shared between the texts. At first glance these clouds may appear to be of the same script, with the consistent repetition of many of the same nouns and verbs. But there are clues that these scripts are different such as references to the Lake Tahoe 'boathouse', Las Vegas 'gambling' and the Cuban hospital 'nurse' from *The Godfather II*. Given enough drafts of a single script it might even be possible, by comparing scriptclouds, to trace a real change of theme between early and later drafts. This in turn could show how the writer's perspective changed as she wrote or how other pressures – genre-conformance, budgetary etc. – necessitated changes in the script.

Compare Content Clouds

[Click here to start a new comparison](#)

The Godfather

anymore **arm** bastard bathroom
bodyguards booths breaks checks
closes coin corridor courtyard crosses
dials disappear drops embrace **enters**
exit falls **family** finishes
gate glances gotta guarantee hangs
hesitates **hospital** hotel hurries
kick kisses knocking leans lifts
limousine louder love luparas **nods**
opens peeks picks **pulls**
pushes refuse relaxes rushes scream
shakes **sits** smiles speeds
stops strength throws tries
walks war wears weeping
whispers wrapped

The Godfather2

accountant airport alley alright **arm**
boathouse bodyguards breaks button
casino **closes** coffin crosses
crowds disappears discussing drives
eats embrace **enters** exits
family fires flashlights
gambling gates glances harbor hotel
immigrant **kisses** knocks lawn leans
lifts lit love **nods** nurse olive
opens owe procession **pulls**
raises realizes rebels recognizes relaxes
rug shakes shouts sings **sits**
smiling statue **stops** strangers
study throws true **walks** war
whispers

Uncommon words

Here is a list of words that are **NOT** shared by the clouds shown above.

anymore, bastard, bathroom, booths, checks, coin, corridor, courtyard, dials, disappear, drops, exit, falls, finishes, gate, gotta, guarantee, hangs, hesitates, hospital, hurries, kick, knocking, limousine, louder, luparas, peeks, picks, pushes, refuse, rushes, scream, smiles, speeds, strength, tries, wears, weeping, wrapped, accountant, airport, alley, alright, boathouse, button, casino, coffin, crowds, disappears, discussing, drives, eats, exits, fires, flashlights, gambling, gates, harbor, immigrant, knocks, lawn, lit, nurse, olive, owe, procession, raises, realizes, rebels, recognizes, rug, shouts, sings, smiling, statue, strangers, study, true

Common words

Here is a list of words that **ARE** shared by the clouds shown above.

arm, bodyguards, breaks, closes, crosses, embrace, enters, family, glances, hotel, kisses, leans, lifts, love, nods, opens, pulls, relaxes, shakes, sits, stops, throws, walks, war, whispers, , arm, bodyguards, breaks, closes, crosses, embrace, enters, family, glances, hotel, kisses, leans, lifts, love, nods, opens, pulls, relaxes, shakes, sits, stops, throws, walks, war, whispers,

Figure 5.19 - Comparing Godfather Scripts

It's also interesting to trace scriptcloud keywords through the 'act' structure of a screenplay or theatre script. In the comparison of the scriptclouds of Shakespeare's Macbeth Acts 1 and 2 below, we can detect some kind of 'turning point' that has changed the atmosphere of the play from Act 1's 'hail', 'honour', 'love' to Act 2's 'bloody', 'daggers' and 'horror'. There is also consistency of language in the 'thee', 'thou' and 'thy' - words that would have been in an Elizabethan stoplist but were not in mine as they do not reflect today's noise words. Further indication of the sensitivity required in

creating an appropriate stop list for the texts to be processed.

Compare Content Clouds

[Click here to start a new comparison](#)

Macbeth Act 1

ambition arm babe battlements blasted
brain castle chamber corporal cries
dearest deed deserves doth drum
earnest false fantastical favour forbid foul
grace greet hail hast hath heath
hereafter hither honour horrid hostess
ignorant ill kinsman lightning love
mortal owe pains praises rapt royal
selfsame shalt sightless sisters tending
thane thee
thine thou thrice thy
torches treasons trifles true vantage weird
withal wonders wouldst wounds

Macbeth Act 2

accents amazed amiss anointed
awaked badged bless bloody
brag castle chamber colour
combustion cried daggers
deed dire distracted
equivocator false fatal feverous
fountain fury grace grooms hath
horror knock livelong
love merciful morrow obscure owl
pillows porter prophesying provokes
renown repent repose royal scales
scream shake sir sore stealing
stirring summons sword tailor temple
thane thee thou thy
treason trusted unwiped vault withal
woeful

Figure 5.20 - Comparing Acts in Shakespeare's Macbeth

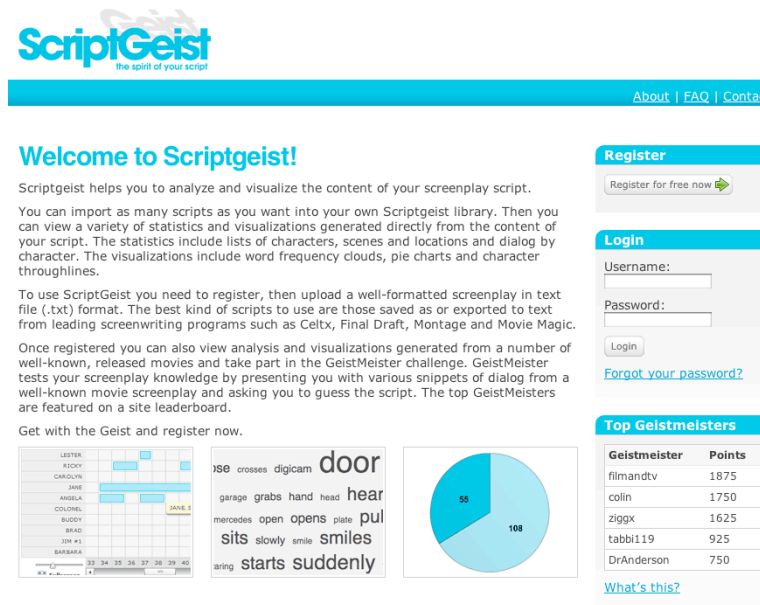
Contentclouds do a useful job of basic 'delta' analysis between two texts to highlight similarities and differences based on the word content only. A potential development of Contentcloud would be to create a scrolling 'timeline' view of drafts where only most frequent 'delta' words between each draft are highlighted as this might help to show how new themes emerge or theme consistency is maintained through the drafts.

5.6 ScriptGeist [2007]

Thanks to funding from LCACE, Scriptgeist.com was created to deliver a better platform for both more, and more sophisticated analyses and visualizations of screenplay content. The aim was to create my first screenplay datafication prototype as, unlike both Scriptcloud and Contentcloud, Scriptgeist accepts a text file upload, parses the content and applies rules to 'shred' it into a set of relational entity data stored in tables in a MySQL database. The main database entities created from the script content are scenes, characters and locations. Scriptgeist is the direct

antecedent of Scenepad and much of what I learned from this prototype is replicated, more or less, in Scenepad.

David North coded the application in Ruby on Rails based on my design and data model. The site is no longer accessible.



© Copyright Stewart McKie 2007. All rights reserved.

Figure 5.21 - Scriptgeist Home Page

Despite offering significantly more functionality than Scriptcloud, the site statistics for Scriptgeist in figure 5.22 show that it only attracted some 89 users who uploaded 71 scripts. Scriptgeist also contained a user engagement tool called Geistmeister to let users review random dialog retrieved from a script to guess the script it came from. Correct guesses got points and the top 5 leaderboard is displayed on the home page (bottom right). Ironically, the Geistmeister challenge proved more popular than the main function of the site itself. Perhaps I had an 'Angry Birds' on my hands and didn't realize it.

Scriptgeist built on Scriptcloud by decomposing the script content into 'snippets' of action or dialog and then generating a series of simple analyses and visualizations, in addition to more scriptclouds, from this content that

provide a useful, if not insightful, foundation for further script content analyses.

Totals		Script totals by genre	
Users	89	Drama	28
User scripts uploaded	71	Comedy	21
Movie scripts uploaded	30	Sci-Fi	14
		Crime	13
		Romance	6
		War	5
		Western	4
		Period	4
		Action	3
		Horror	3
		Road	0
		Spy	0
		Bio	0
		Fantasy	0

View counts for visualisations	
Script	400
Character Thruline	299
Characters	296
Locations	212
Scenes	202
ScriptCloud	191
DialogCloud	173
Day vs Night	155
Interior vs Exterior	150
ActionCloud	133

Figure 5.22 - Scriptgeist Statistics

Scriptgeist enables a user to generate three keyword clouds based on all the script text (as in Scriptcloud), or clouds based on dialog or action text only.

ScriptGeist Administration

Logged in as [cfoinfo](#) | [Log out](#) | [View website](#)

Stats Static pages FAQ Database Users Settings Stop words Messages

Stats

You are here: [Your Scripts](#) » [Godfather, The](#) » [DialogCloud](#)

Godfather, The - DialogCloud

The most frequently used words in the dialog text.

[Sort by frequency](#)

alive america american anymore bastard blood boss boy brasi brother
business calling christ consigliere **corleone don**
 eh families **family** freddie **godfather** gonna gotta
 guarantee guy hospital love mall **mike** pop refuse santino shot sicily
 son tattaglias ten thousand **tom** tomorrow true turk understand war
 wedding wife wine wish york young

[Download as pdf](#)

[Add this cloud to your website/blog](#)

Godfather, The

[select another script](#)

[Download original file](#)

Visualizations

- Script
 - ScriptCloud
 - DialogCloud
 - ActionCloud
 - Character Thruline
 - Interior vs Exterior
 - Day vs Night
- Scenes
- Characters
- Locations

© Copyright Stewart McKie 2007. All rights reserved.

Figure 5.23 - The Godfather Dialog Cloud

Scriptgeist generates a character throughline chart to show which scenes the top 10 characters (by dialog quantity) appear in throughout the script.

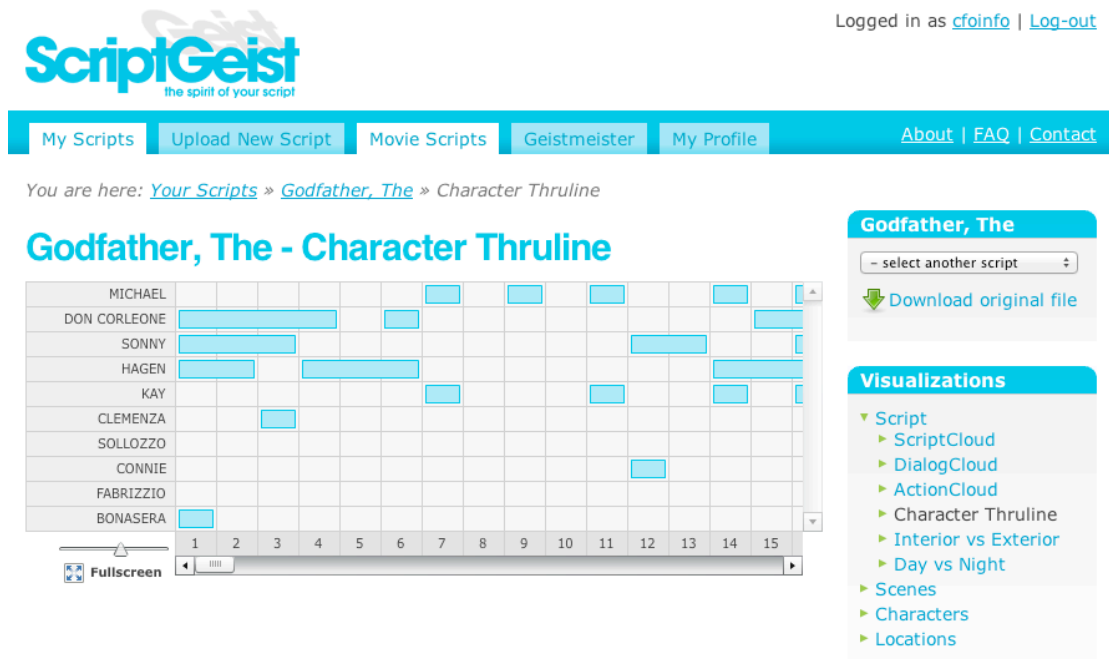


Figure 5.24 - The Godfather Character Throughline

Scriptgeist generates pie charts to show the scenes in a script categorized by INT/EXT and DAY/NIGHT.



© Copyright Stewart McKie 2007. All rights reserved.

Figure 5.25 - *The Godfather* INT/EXT Scenes

A list of scenes, characters and locations are generated from the uploaded script text. The character list shows the number of words and dialog snippets linked to a character to help analyze comparative part-size.

ScriptGeist
the spirit of your script

Logged in as [cfoinfo](#) | [Log-out](#)

My Scripts | [Upload New Script](#) | [Movie Scripts](#) | [Geistmeister](#) | [My Profile](#) | [About](#) | [FAQ](#) | [Contact](#)

You are here: [Your Scripts](#) » [Godfather, The](#) » Characters

Godfather, The - Characters

This script contains 51 characters

Name	Words	Pieces of dialog	Main character?
Michael	2365	210	No
Don corleone	2082	104	No
Sonny	1244	96	No
Hagen	1342	93	No
Kay	415	64	No
Clemenza	770	46	No
Sollozzo	629	35	No
Connie	293	20	No
Bonasera	336	18	No
Fabrizio	214	18	No
Tessio	272	17	No
Fredo	155	16	No
Carlo	110	16	No
Paulie	91	12	No
Mama	138	11	No
Don tomassino	164	10	No
Moe	208	10	No
Vitelli	74	9	No
Mccluskey	170	9	No
Woltz	321	7	No

« Previous 1 2 3 Next »

© Copyright Stewart McKie 2007. All rights reserved.

Godfather, The

— select another script —

[Download original file](#)

Visualizations

- ▶ Script
- ▶ Scenes
- ▼ Characters
- ▶ Locations

Figure 5.26 - *The Godfather* Characters and Part Size

Clicking on the dialog snippets number lists all dialog text spoken by a character to help analyze voice consistency.

My Scripts | Upload New Script | Movie Scripts | Geistmeister | My Profile | About | FAQ | Contact

You are here: [Your Scripts](#) » [Godfather, The](#) » [Characters](#) » Michael

Godfather, The - Characters

Michael
[Details](#) | [Cloud](#) | [Dialog](#)

Filter dialog:

I told you I had a lot of relatives.	<input type="checkbox"/> Main?
They're waiting to see my father.	<input type="checkbox"/> Main?
They're going to talk to my father, which means they're going to ask him for something, which means they better get it right.	<input checked="" type="checkbox"/> Main?
Because they know that no Sicilian will refuse a request on his daughter's wedding day.	<input type="checkbox"/> Main?
No. His name is Luca Brasi. You wouldn't like him.	<input type="checkbox"/> Main?

Godfather, The

- select another script ▾

Download original file

Visualizations

- ▶ Script
- ▶ Scenes
- ▶ Characters
- ▶ Locations

Figure 5.27 - The Godfather Character Dialog

Scriptgeist delivered a wide range of screenplay analytics functionality and only lacked the ability to write and manage a script and to share a script for social networking purposes. At the time, to my knowledge, it was the only application of its kind that was publicly available.

5.7 SceneClass [2010]

Given the explosion of the blogging paradigm and pervasive use of social networks by 2010, it seemed important to explore how to socialize the scene writing process. I decided to do this by leveraging the popular, open source blogging platform Wordpress as the basis for my next prototype called SceneClass.com. The aim of SceneClass was to focus on scenewriting as a group activity where the scene content functioned like a blog post.

The application was coded by Nick Ohrn and deployed as a Wordpress plugin. Apart from John August's *Scrippets*, a text formatting plugin for generating screenplay-formatted text from plain text input, this was the first group screenwriting plugin ever produced for Wordpress. The site is currently located at: <http://sceneClass.tripos.biz>.

Note that some of the screenshots below obscure the names of the participants for privacy reasons.

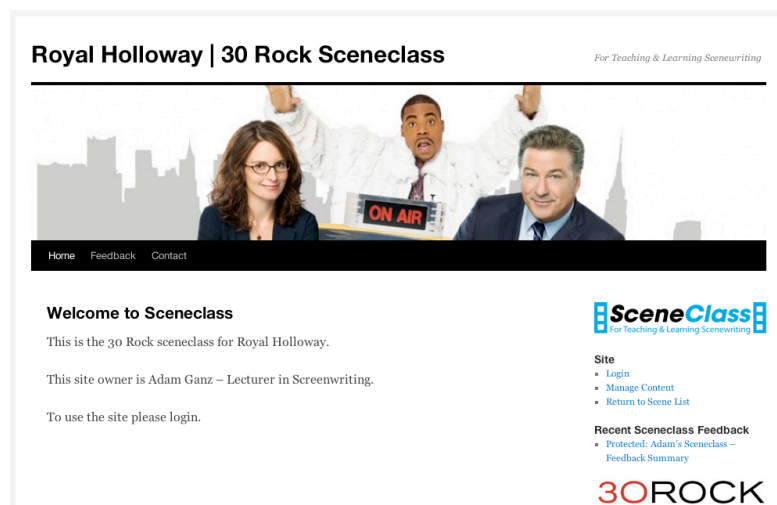


Figure 5.28 - SceneClass Home Page

SceneClass was tested in July 2010 by a group of 12 students at a scenewriting class tutored by Adam Ganz at Royal Holloway as part of a Masters level course in screenwriting run by Sue Clayton. The task was to write a 'cold opening' for the hit US comedy series *30 Rock*. In a period of about 3 hours the students not only got up to speed on SceneClass, but also wrote a scene and contributed to the 88 comments posted to the scenes via SceneClass. Feedback from students on the sceneclass paradigm was positive and they had no difficulty using the prototype following a brief 'getting started' presentation on the day.

The SceneClass paradigm starts with a Wordpress user group called a 'Class' with two user roles - a 'Tutor' and a 'Student' - each with a set of rights. For example, only a tutor can create a class.

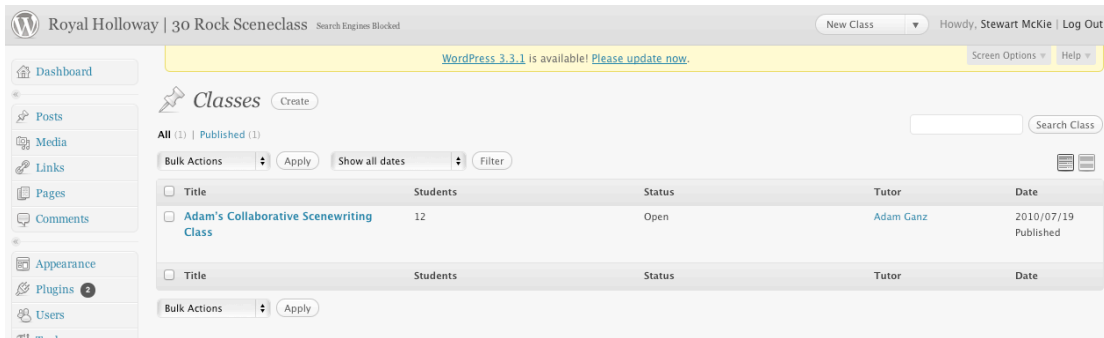


Figure 5.29 – SceneClass Class Entity

The tutor or the students create a ‘Script’ as a container for ‘written by the students. Students could create their own script and scenes to share with the group or the tutor could create a single ‘group script’ into which students in the group add their own scenes.

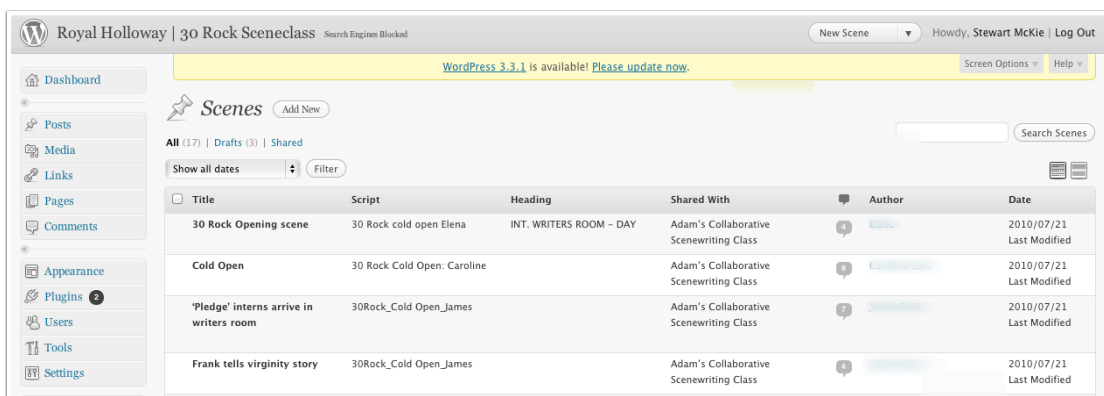


Figure 5.30 - SceneClass Scene Entities

Scenes are written just like a regular Wordpress blog post [explain] but with certain conventions e.g. character names are in CAPS so that the scene text can be formatted to screenplay conventions. Once a scene is ‘posted’ to the script it can be viewed correctly formatted. The scenes are flagged as ‘private’ or ‘shared’ to share with the rest of the group so that each scene may be commented on by the rest of the class in much the same way as any regular blog post in Wordpress (or any other blog).

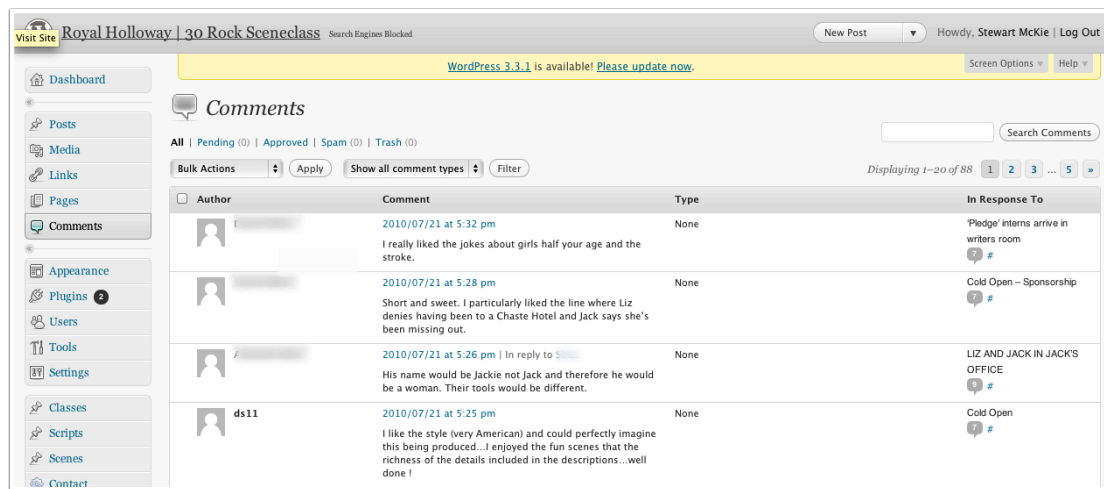


Figure 5.31 – SceneClass Scene Comments

Properly formatted scenes can be viewed by anyone in the class group and commented on to provide a social scenewriting experience, as shown below.

The blogging paradigm is quite suitable for scene writing in that each blog post functions as a scene and a collection of posts represent a script. With a little background work in CSS, scenes can be viewed correctly formatted and commented on, shared and rated/'liked' just like a regular blog post.

Rewriting is easy, you just edit your post, and scenes can be tagged to aid with searching and the wealth of Wordpress plugins includes word cloud generators that could create a scriptcloud from a series 'scene posts' either by a single writer or a class of students.

4 Responses to *30 Rock* Opening scene



..... says:

July 21, 2010 at 5:10 pm

like that Jack himself is so excited about having virgins around he keeps muddling up their titles and calling them virgins instead of interns.

[Reply](#)



ds3 says:

July 21, 2010 at 5:16 pm

Very funny! I think you got a good hold of the characters, very believable.

[Reply](#)



..... says:

July 21, 2010 at 5:20 pm

Love the two beats of virgin mistake from Jack. Maybe better from Liz? Very funny.

[Reply](#)



..... says:

July 21, 2010 at 5:23 pm

Love the slip of the tongue Jack has. Good set-up.

[Reply](#)

Figure 5.32 – Sceneclass Student Responses

Sceneclass showed that students quickly ‘got’ how to use the scenewriting and commenting paradigm and in the post-session feedback, they said they got value from it. A handful of other Universities and colleges expressed an interest in using Sceneclass and some tried to use it but it failed to gain any real traction with these prospective users. Despite this poor take-up, commenting is an integral part of Scenepad and is informed by my experience of developing and using Sceneclass.

Summary

A number of online deliverables were made publicly available to prototype new or different ways of managing and socializing, analyzing and visualizing screenplay content. All of these prototypes were used, more or less, by a range of users including practitioners, academics, school, undergraduate and postgraduate students. Although all the prototypes included ways to contact me or provide feedback and in some cases formal feedback was requested, actual feedback was minimal and so their research value limited.

These days, the availability of over a dozen free online screenwriting applications competing for user mindshare means that many users (myself included) tend to sign up, try the app and then never return. This was the case with my prototypes – for example in Scriptcloud most users who signed up then uploaded a script. But once they had generated a scriptcloud from their script they never came back. Scriptgeist users came back mainly to play the Geistmeister game and prospective SceneClass users could not put enough time and effort into getting SceneClass running in their institutions despite good intentions.

So on the basis of the user response to these prototypes and the lack of substantive feedback, it is difficult to draw any valid conclusions about the usefulness or not of the functionality provided.

6 Scenepad

Scenepad is my final 'assessment' deliverable. It is informed by the prototypes that preceded it, as discussed above. Scenepad is my attempt to distil some of the best of these prototypes into a single application that can be used to create and manage, present and share, analyze and visualize, screenplay content in new or different ways. It is my practical contribution to the screenwriting 2.0 debate and I hope that much of what Scenepad delivers will become standard functionality in the next generation of commercial screenwriting applications.

Scenepad was coded (at various times) in PHP & Javascript by myself, Scott Darby, Matt Kenyon and Hannah King based on my design specifications and my data model in MySQL. Scenepad is owned by Tripos Publishing Ltd., the company that funded the occasional costs of the coding effort.

The first iteration of Scenepad ('scenepad1') went into a limited public 'beta' phase in January 2012 at this location: <http://scenepad.tripos.biz>

Scenepad2 iteration can be found at: <http://scenepad2.tripos.biz>

Scenepad3 iteration can be found at: <http://scenepad3.tripos.biz>

Scenepad4, my final submission iteration, can be found at:
<http://scenepad4.tripos.biz>

A full list of contributors to Scenepad4 and components used in the application can be found on the site's Credits page at:
<http://scenepad4.tripos.biz/credits.php>.

The screenshots provided in this section reflect all 4 iterations of Scenepad, each of which had a slightly different user interface. The second to fourth iterations of Scenepad use the popular Twitter Bootstrap framework for the user interface so look slightly different from the initial version. The third and fourth iterations use examples of a 'responsive' user interface, better for use

on an iPad as it resizes automatically. These four iterations are referred to as Scenepad1, 2, 3 and 4 below.

The Scenepad1 beta was tried by a small group 'beta test' users of which only a couple of user actually engaged usefully with the prototype by providing a few comments in the application and some email feedback. I wish to thank Carmen Sofia Brenes of the Universidad de los Andes for her help in this beta test. As a work in progress subject to multiple user interface and functionality changes, it was perhaps unrealistic to expect busy academics to engage with testing a screenwriting application like Scenepad.

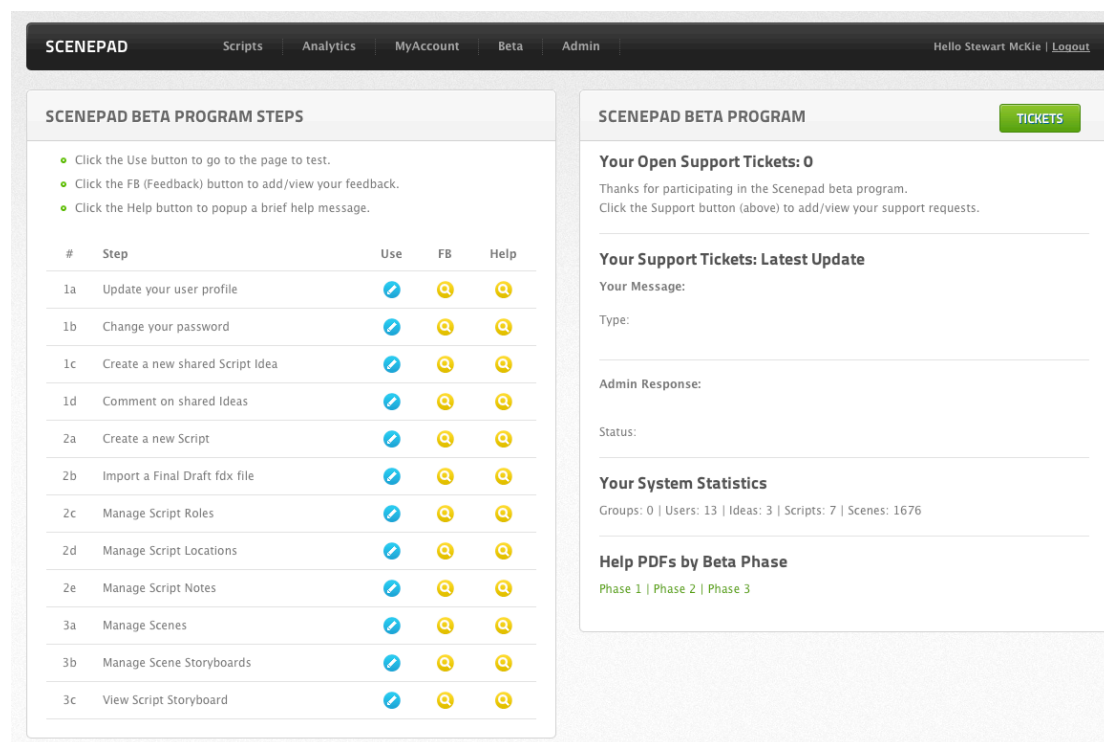


Figure 6.1 - Original Scenepad1 Beta Home Page

6.1 Scenepad – Screenplay as Data

All Scenepad content is stored as data and metadata in tables in a MySQL relational database management system (RDBMS). So, for example, the PDF files that can be downloaded to print out script or scene content are generated on-the-fly by querying data and metadata in the database tables and do not pre-exist as files stored in the file system. Any screenplay textual

data (e.g. dialog or action snippet text) is not stored as a text file but as data in a column in a table.

Scenepad clearly has a 'database' rather than 'document' design and delivery paradigm. This will not suit many screenwriters, maybe most screenwriters. But this approach has enabled me to produce an application that is fully featured and runs entirely online for relatively low cost. I chose this approach as the basis for Scenepad largely because I understand SQL databases better than the XML file format, which would have been the obvious alternative choice as a way to datafy the screenplay content.

In MySQL, the Scenepad design is anchored by three main tables: script, scene and snippet and these main entities are supported by role and location tables. A 'snippet' is a block of action description, dialog, a caption or a transition. Snippets belong to scenes and scenes belong to scripts. A script must contain one or more scenes and a scene must contain one or more snippets. From a production perspective, a snippet could be linked to one or more shots. The current version of Scenepad does not support shots as an entity concept or linking shots to snippets/scenes as indicated by the dashed arrow & box, but this would be relatively easy to add (although low-cost applications like ShotList (see <http://solubleapps.com/shotlist/>) already do a very good job of this on iPhone devices).

Scenes take place at locations and scene dialog snippets are spoken by a role. The actual content of a scene is stored in two tables: the snippet table (representing the current 'datafied' version of the scene content) and the version table, which stores the text of current and all previous versions that were saved to the database.

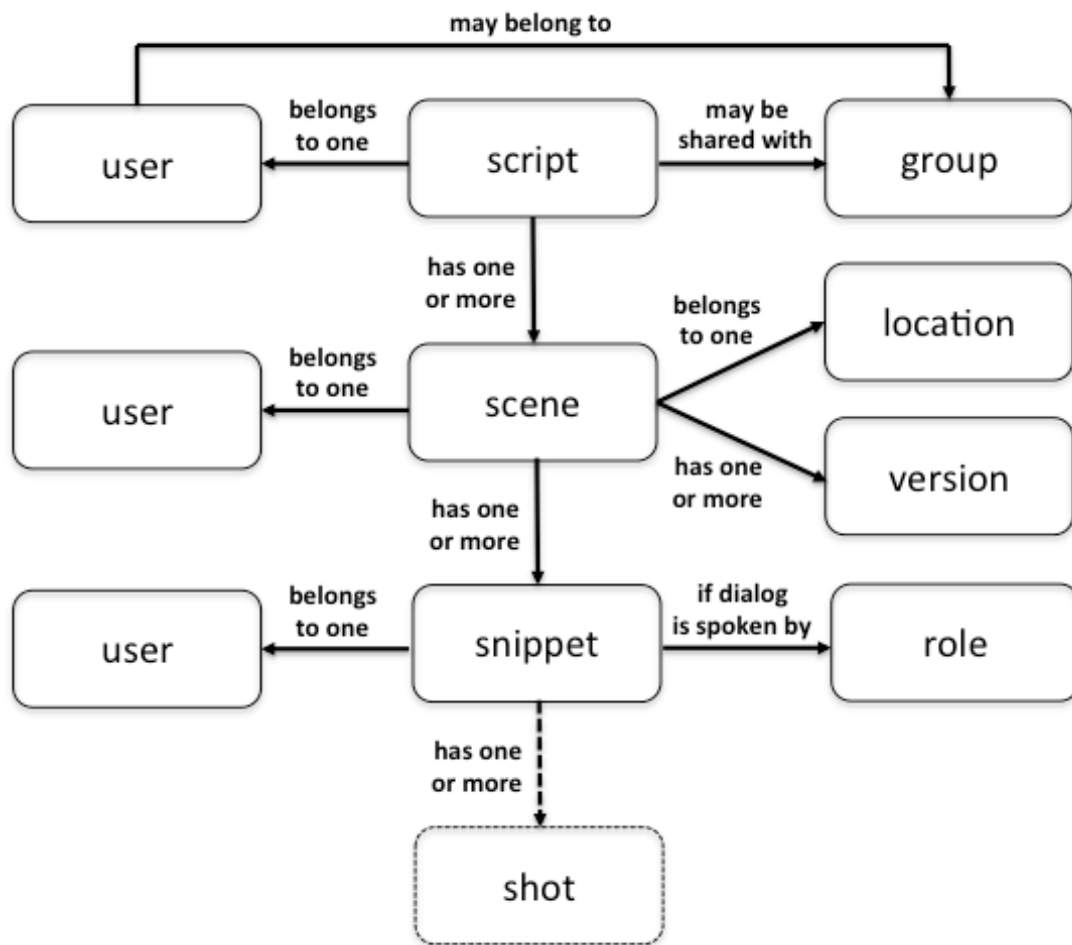


Figure 6.2 – Core Scenepad Data Entities (Tables)

The datafication of the script in this way means that any kind of presentation of the script content, from an individual snippet to the whole script, is in fact based on a database query. Scenepad does not retrieve a file containing the script content but assembles the script content on the fly using a query depending on what you want to see and how you want to see it. This is a different way of interacting with a screenplay than the traditional document-focused presentation. Scenepad provides a PDF version of the script at multiple levels: Script, Role, and Location. The PDFs include all the scenes that are linked to the whole script, role or location. Sequence and storyline level PDFs are also available.

Screenplays stored as document files are usually very easy to search to find specific instances of a term. Scenepad also includes 3 search pages to search

all snippet text or only dialog snippets or action snippets.

6.2 Scenepad - Scenewriting

Traditional document-centric screenwriting software is focused on writing scripts. Scenepad is focused on writing scenes. In Scenepad, the script acts merely as a container for scenes. The primary entity that the user interacts with to actually write a Scenepad script, is the scene. This means that individual scenes can be shared, rather having to share a whole script and an individual scene can be versioned and compared, which facilitates collaborative working and could help with the activity of 'polishing' a script that often focuses on specific scenes or script dialog content only.

Given Syd Field's assertion that 'The Scene is the single most important element in your screenplay' (2005:162), screenwriter David Mamet's definition of a film as essentially a progression of scenes (2007:85) and Wells (2011:101) perspective of scenes as a micro-narrative in their own right (in animation scripts), surprisingly few screenwriting pundits and academics have focused on scenewriting as a discipline. Any significant reference to the art of scenewriting is also missing from works focused on screenplay 'coverage' such as Sher's *Reading Screenplays* and Garfinkel's *Screenplay Story Analysis*.

McKee considers scenes to be comprised of a series of 'beats' and scenes themselves to form the beats of sequences: 'Ideally every scene is a STORY EVENT' (1998:35-38). Scofield defines a scene as having four basic elements: event and emotion, function, structure and a pulse. (2007:14). Aronson emphasizes the importance of the scene that includes the first turning point as the one *striking and unpredictable scene* (2010:101) and what the audience needs to know about characters in a scene (ibid:93). Batty and Waldeback include a useful section on scenes as an aspect of structure and narrative, emphasizing the need for scenes to have their own arc, the importance of topping and tailing a scene, and paying attention to scene transitions in relation to their preceding and succeeding scenes,

commenting (2008:39-47):

Much scene work tends to be intuitive but it is very useful, not only for writers but for all development personnel, to analyze and approach them with a committed level of craft.

Sternberg (1997:65-66) analyzes scenes by dividing scene content into dialogue text and scene text, which roughly correspond to Scenepad's dialog and action snippet types except that Sternberg considers that certain other 'instructions' about either a snippet or a scene e.g. character cues, transition instructions and scene headings to be part of the scene text rather than metadata that refers either to scene or snippet text.

Scene Versioning

Most scriptwriting applications also lack the ability to version scenes and compare versions. One exception is Mariner's *Storymill* application that does include the ability to compare a current to a prior version (or 'snapshot' as they call it) and to replace the current version with a prior version or delete versions.

There are a number of reasons why scene versioning can help scriptwriters, especially with the rewriting process:

1. To track the development of a scene over time.
2. To easily return to a previous version of a scene to replace the current version.
3. To compare different versions of the same scene written by different people e.g. within a writing team or a screenwriting class.
4. To 'cherrypick' the best text from alternative versions and aggregate it into a current version.

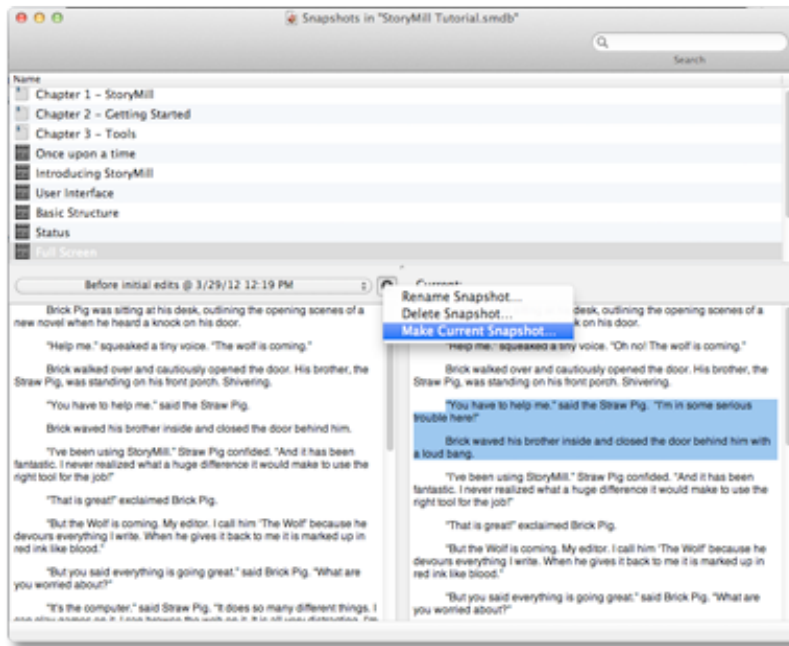


Figure 6.3 - Mariner *Storymill* Scene Snapshots

Scenepad1 allows comparison of the current version against any number of previous versions and highlights the differences between the current and 'reference' version – red for deleted words and green for inserted words.

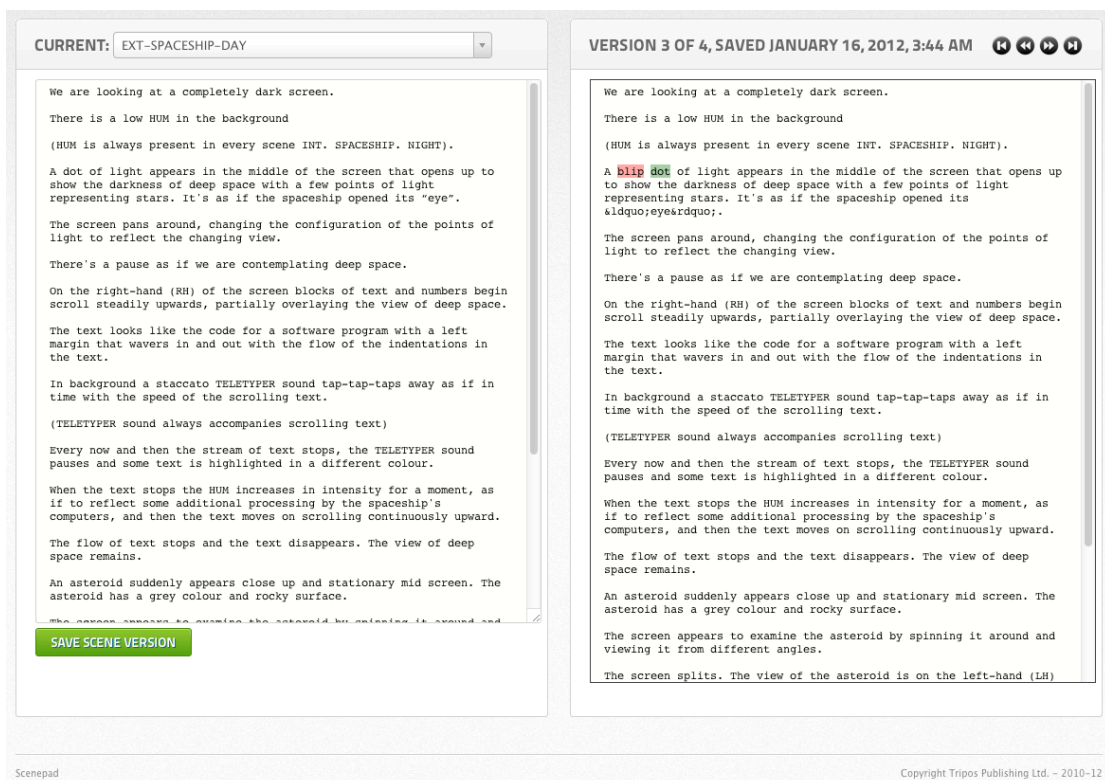


Figure 6.4 - *Scenepad1* Scene Versioning

Scene Grouping

Scenepad3 & 4 include the capability to group scenes into dramatic units or by specific types of scene. Scenes can be linked to either a storyline or a sequence or flagged as having a specific content e.g. SFX, VFX, Sexual, Violence, Crowds or stunts. Flagging scenes in this way enables storyline and sequence thrulines to be visualized and a set of dials used to quickly understand the how profane, violent or stunt heavy, for example, a script is written to be (see figure 6.5). Scene status can also be indicated with a 'traffic light' to show writing status e.g. in-play (amber) or final (green) in the scene list (see figure 6.6).

RECONCILIATION Scene Types

Use Scenes>Manage to edit scene data to flag scenes with special types. Dials only update if enough scenes are flagged with types to register.

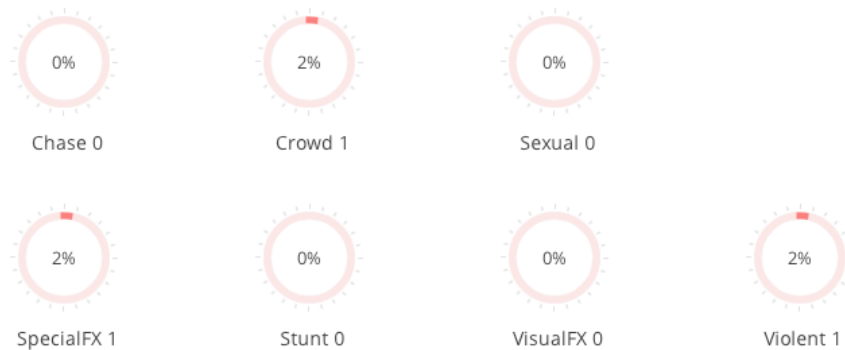


Figure 6.5 – Scenepad3 – Scene Content Types and Analysis

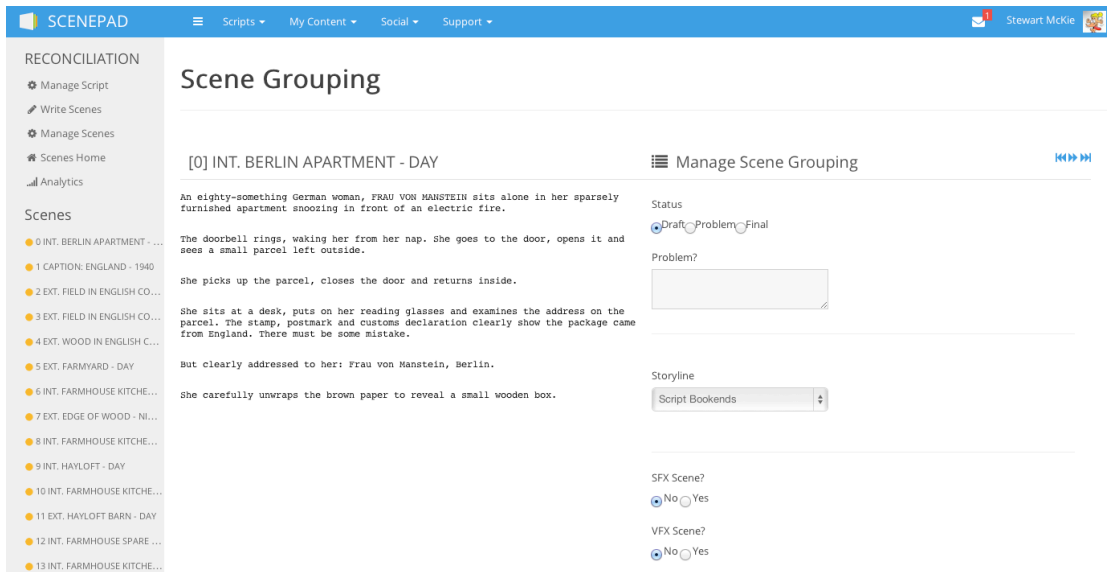


Figure 6.6 – Scenepad4 Managing Scene Grouping

Scene Structure Template

Scenepad3 & 4 also enable writers to quickly apply the basics of Vogler’s Hero’s Journey as a template for their script by optionally selecting to use the template when the initial script entity is created. This ‘starter’ template automatically creates the 12 core sequences and 8 character archetypes for the writer to build on.

The screenshot shows the Scenepad3 interface for a script titled 'FDX Test Script'. The left sidebar contains navigation options like 'My Scripts', 'Import Script', 'Shared with Me', 'Public Scripts', 'Scenes', 'Roles', 'Locations', 'Analytics', 'Clouds', 'Thrulines', 'Structure', and 'Business'. The main content area displays two tables related to a 'Hero Journey'.

Hero Journey Sequences

#	Act	Seq	HJ Stage	Your Sequence Name				
1	Act1	1	Ordinary World	Ordinary World				
2	Act1	2	Call to Adventure	Call to Adventure				
3	Act1	3	Refusal of the Call	Refusal of the Call				
4	Act1	4	Meeting with Mentor	Meeting with Mentor				
5	Act1	6	Crossing Threshold	Crossing Threshold				
6	Act2	6	Road of Trials	Road of Trials				
7	Act2	7	Approach to Cave	Approach to Cave				
8	Act2	8	Ordeal	Ordeal				
9	Act2	9	Reward	Reward				
10	Act3	10	Road Back	Road Back				
11	Act3	11	Resurrection	Resurrection				
12	Act3	12	Return with Elixir	Return with Elixir				

Hero Journey Roles

#	1 Cue	Logline	Gender	HJ Archetype	Role Type	Thruline				
1	ALLY	-None-	Male	Ally	Main	N				
2	GUARDIAN	-None-	Male	Guardian	Main	N				
3	HERALD	-None-	Male	Herald	Main	N				
4	HERO	-None-	Male	Hero	Protagonist	N				
5	MENTOR	-None-	Male	Mentor	Main	N				

Figure 6.7 - Scenepad3 Hero's Journey Template

Scene Sequencing and Storylining

Scene sequencing, grouping scenes into dramatic units called 'sequences', is missing from both Final Draft and Celtx. While both include scene navigators to jump around a script and focus on a specific scene, with Celtx providing a nice drag-and-drop feature for reordering scenes on the fly, neither formally recognize either the 3-Act structure or grouping scenes either into sequences or plot/storylines. Scenepad 3&4 provides a drag and drop scene reordering while Scenepad1 provided a script navigator to navigate through scenes by means of Acts and Sequences using a conventional collapse and expand tree (see figure 6.8 below).

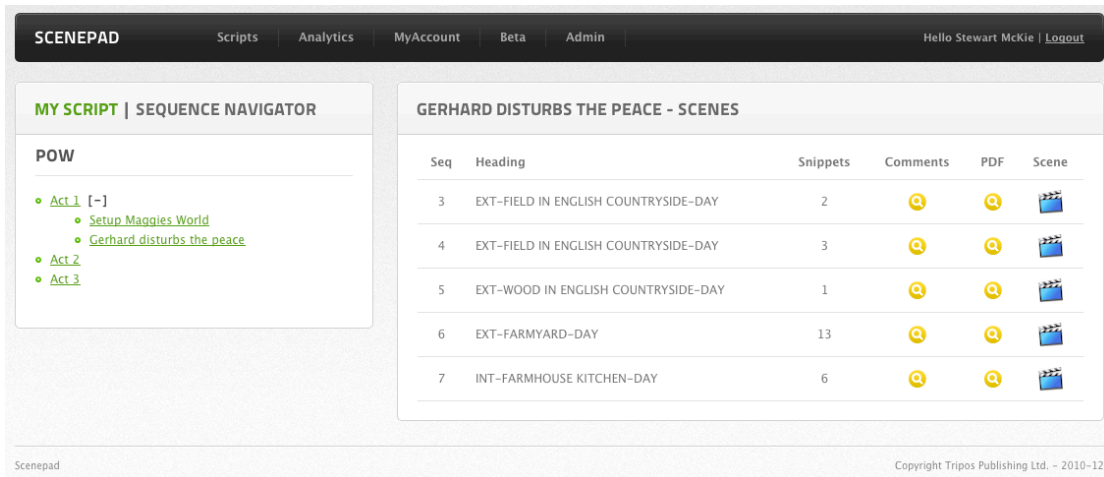


Figure 6.8 – Scenepad1 Act and Sequence Navigator

Scenepad4 lets writers tag scenes as 'belonging' to a sequence or a storyline (see figure 6.9). This enables the script to be managed by sequence or storyline so that, for example, it would be possible for a specific user or user role to manage one or more sequences or storylines to effect a division of labour within a writing team. Scenepad4 also lets you view scenes or print a PDF of the scenes by sequence or storyline so the view/PDF only includes scenes tagged to that specific sequence or storyline to help review and manage these structural entities more easily.

Scene Storylines

#	↑ Name	Logline	Active	Scenes	
1	Doreen & Sergeant	Unwitting antagonists to Maggie	Yes	0	📄 🎬 🔍 📝 ✖
2	Gerhard Alone	Gerhard before and after Maggie	Yes	4	📄 🎬 🔍 📝 ✖
3	Maggie & Doreen	A conflict relationship	Yes	0	📄 🎬 🔍 📝 ✖
4	Maggie & Gerhard	The core relationship	Yes	1	📄 🎬 🔍 📝 ✖
5	Maggie & Harold	The relationship between Maggie and Harold the...	Yes	1	📄 🎬 🔍 📝 ✖
6	Maggie & Sergeant	A conflict relationship	Yes	0	📄 🎬 🔍 📝 ✖
7	Maggie Alone	Maggie on her own	Yes	3	📄 🎬 🔍 📝 ✖
8	Script Bookends	Start and finish bookending scenes	Yes	1	📄 🎬 🔍 📝 ✖

Figure 6.9 – Scenepad4 Scene Storyline Manager

Scene Storyboard

Scenepad4 includes a simple storyboard view for attaching images to a

scene. These images could be traditional hand-drawn boards or images found on the Internet – as is the case in figure 6.10. Given that Scenepad is multi-user and role-based it would also be possible to create a ‘storyboarder’ role that only has access to scene text and the storyboarding functions so this task in the script development process can be easily ‘farmed out’ to an external expert.

📁 RECONCILIATION

EXT. FIELD IN ENGLISH COUNTRYSIDE - DAY8

[Manage Storyboard](#)

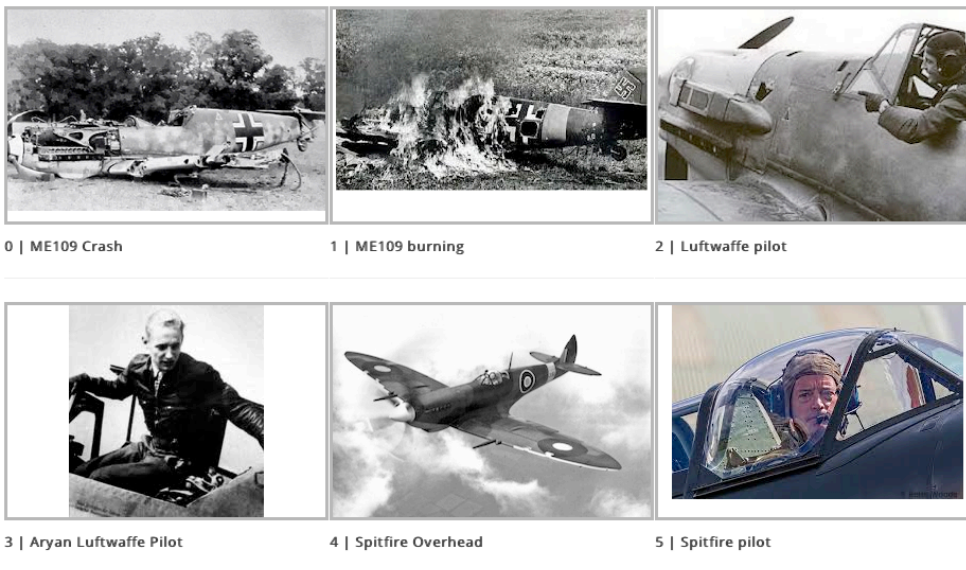


Figure 6.10 – *Scenepad4* Scene Storyboard

6.3 Scenepad as Social Network

Scenepad recognizes that it may be advantageous to the writer or simply necessary to share a script with others. Scenepad enables the script owner to decide the sharing status of her script. The script status ‘private’ means script content is only accessible to the script ‘owner’ user. The script status ‘shared’ means script content is may be accessible to another user or other users that are part of a group that the script owner creates and invites users to join. Scenepad groups (see figure 6.11) could reflect a screenwriting class at a college, a writing team or the members of a production set for example.

The script status 'public' means the script content can be viewed (only) by any Scenepad user. This is useful, for example, to provide 'reference' screenplays to a scriptwriting class.

SCENEPAD Scripts My Content Social Support Stewart McKie

Script Community

A community is a group of users to share scripts with. A community could be a writing team, a screenwriting class or production set staff.
Click Add to add a new community. Click the user icon to add users to the group via an email invitation.

+ Add Print

#	Type	↑ Name	Users	
1	Class	Scenepad Beta Group	23	
2	Set	Scenepad Production Set	0	
3	Team	Scenepad Writing team	0	

Figure 6.11 – Scenepad4 Groups

Group members have a role within the group and may have the sharing rights of 'Partner' or 'Viewer' – the former enabling more content addition and editing rights than the latter (see figure 6.12).

SCENEPAD Scripts My Content Social Support Stewart McKie

Scenepad Beta Group Users

A user is a member of a Script community. The user may have Viewer or Partner level sharing rights.
A Viewer may view script content, comment and add content attachments. A Partner may add or edit scene content.
Click Add to add a new user to the current community. The user will be sent an email invitation to confirm their Scenepad account.

+ Add Print

#	First	↑ Last	Company	Email	Group Role	Rights	Active	
1	Marie	Amsler		Send	Writer	Viewer	Yes	
2	Carmen Sofia	Brenes	Universidad de los Andes	Send	Writer	Viewer	Yes	
3	Scott	Darby	Scott Darby	Send	Writer	Partner	Yes	
4	Adam	Ganz		Send	Writer	Partner	Yes	
5	Andrew Kenneth	Gay		Send	Writer	Viewer	Yes	

Figure 6.12 - Scenepad4 Group Users

Individual scripts can be shared with a specific group so all users within the group can engage with the script.

Screenplay as Content Hub

If you accept that a script has a creative lifecycle then you should also accept that it has the potential to have a 'scriptstream' – a variant of the 'lifestream' concept conceived by Eric T Freeman at Yale University in 1997:

A lifestream is a time-ordered stream of documents that functions as a diary of your electronic life; every document you create and every document other people send you is stored in your lifestream. The tail of your stream contains documents from the past (starting with your electronic birth certificate). Moving away from the tail and toward the present, your stream contains more recent documents --- papers in progress or new electronic mail; other documents (pictures, correspondence, bills, movies, voice mail, software) are stored in between. Moving beyond the present and into the future, the stream contains documents you will need: reminders, calendar items, and to-do lists.

Replace 'life' with 'script' and 'your' with 'its' and you have the concept of a 'scriptstream' that in turn helps to reframe screenwriting as a process, and the screenplay as a content hub.

This scriptstream content may take the form of images, location videos, voice clips and continuity snaps. All of which serve the dual purposes of both helping to realize the script in a professional manner and providing a record of the realization process itself for future diagnosis and analysis say in the editing process or to assist with filming reshoots or pick up shots.

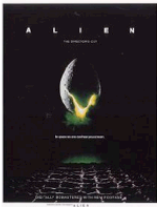
The scriptstream provides rich content that annotates the script itself and can significantly enhance its meaning and immersion of workgroup roles in the 'world' of the script. There are at least three ways this content can be added to the script.

1. Writers could sit at a desk and attach content found on the Internet.

2. Cast and crew 'on set' could capture content as it is produced.
3. Cast and crew 'on location' or simply 'out in the world' can capture content e.g. photos that they think will enhance the realization process.

In the case of [2] and [3] above, this depends on the use of mobile devices as content capture devices and then on the ability, somehow, to link content captured 'on the go' to the relevant aspects of the script – for example to link a video of a building to a location in the script or a recording of a birdsong in a forest glade to the soundtrack of a scene. Scenepad4 provides a content hub at script level (see figure 6.13) that enables users to view almost all data associated with a script in one place and to link social content in the form of comments, documents, images and web links.

Alien | Dan O'Bannon



[Mobile Site Link](#)

[IMDB Link](#)

In space, no-one can hear you scream

The crew of a commercial deep space mining ship, investigating a suspected S.O.S., lands on a distant planet and discovers a nest of strange eggs.

The small crew and confined spaces mean that most scenes have an intimate, close quarter feel to them.

As the script progresses and the Alien kills off the crew, everything focuses more and more on the protagonist, Ripley.

Apart from a few scenes on the alien planet surface, all the action takes place in locations within the spaceship Nostromo.

Content Stats

- 264 Scenes
- 10 Roles
- 69 Locations
- 2 Comments
- 0 Documents
- 1 Images
- 2 Links

Script Content

Scenes
Roles
Locations
Comments
Documents
Images
Links
Tweets
Keeps

Scenes

[Print](#) [CSV Export](#)

#	↑ Order	Heading	Snippets	Comments	+ Ratings	
1	1	INT. ENGINE ROOM	1	1	1	📄 💬 🔍
2	2	INT. ENGINE CUBICLE	1	0	0	📄 💬 🔍
3	3	INT. OILY CORRIDOR - "C" LEVEL	1	0	0	📄 💬 🔍
4	4	INT. CORRIDOR - "A" LEVEL	1	0	0	📄 💬 🔍
5	5	INT. INFIRMARY - "A" LEVEL	1	0	0	📄 💬 🔍
6	6	INT. CORRIDOR TO BRIDGE - "A" LEVEL	1	0	0	📄 💬 🔍
7	7	INT. BRIDGE	3	0	0	📄 💬 🔍
8	8	INT. CORRIDOR TO HYPERSLEEP VAULT	1	0	0	📄 💬 🔍
9	9	INT. HYPERSLEEP VAULT	1	0	0	📄 💬 🔍
10	10	INT. GALLEY	2	0	0	📄 💬 🔍

10 ↓
1
2
3
4
5
6
7
...
26
27
Search

Figure 6.13 - Scenepad4 Script Manager Content Hub

Scenepad4 also provides a content hub concept at scene, role and location entity levels. We have already seen above how a scene can have a storyboard attached to it. Scenepad also assumes that both roles and locations may also have their own gallery of images attached to them to act as further visualizations of the 'world' of the script (e.g. role gallery in figure 6.14). These galleries do not function as a storyboard in that the images are not intended to be sequenced (i.e. as individual 'shots' are in a scene storyboard) instead they function as a place to gather visual research, say about the character, and to visually remind the writer of the character they are writing for. A location or role gallery can also be useful to propose the kind of location or character that is in the writer's mind.

RECONCILIATION

Role Gallery

[Manage Gallery](#)






		
0 SERGEANT Oliver Reed (RIP) - Triple Echo	1 MAGGIE Jenny Agutter?	2 MAGGIE Susan George?
		
3 MAGGIE Catherine McCormack?	4 MAGGIE Glenda Jackson-Triple Echo	

Figure 6.14 - Scenepad4 Role Gallery

Script Mashups

Once the screenplay becomes a content hub then the need for 'mashups'

becomes more likely. Here, the term mashup is used to reflect the ability to use application programming interfaces (APIs) to link a screenwriting application to other online applications in order to combine content or provide specialist functionality provided by the third party application.

Scenepad1 included a mashup with Dropbox - a popular online file storage service - that facilitates its ability to support the Screenplay as content hub paradigm. Scenepad allows script, scene, role and location entities to have any number of attachments attached to them. The simple way to do this is simply to link to the content via a web link. But by linking Scenepad to Dropbox, users can retain control over their own attachment content, by physically storing it in their Dropbox account, and Scenepad can leave the storage and management of this content to a service that is optimized for this purpose. Neither Final Draft nor Celtx have any support for mashups of this kind. Scenepad2 included a mashup with blogging platform Tumblr to link content posted to Tumblr with a Scenepad script. Scenepad3/4 enables users to view photos posted by email to kee.ps and to view Tweets posted to a specific Twitter user account (which could be the name of a script for example).

In the script realization stage (i.e. during production), writers, cast and crew adopt the generic role of 'content-providers' to the script, each with a specific focus in terms of the content they provide and the script entities that they link it to. Now the script can become the 'hub' of a universe of content that helps to realize and reflect the world of the script realization process. And surrounding the script with this content may begin before any actual filming has taken place as part of the pre-production activity. Then, as a result of this additional linked content, a script is realized in two ways, both as an intermediary 'rich-content' artefact and as some form of end product – the audio/visual experience itself.

Scene Commenting

Scenepad4, like the SceneClass prototype before it, recognizes that scenes

are the most likely entity to benefit from social interaction through commenting (cf. SceneClass above). So the scene navigator includes the entry and viewing of 4 types of comment: A general comment, a highlight, a question and an idea or suggestion as shown in figure 6.15 below. The scene text is displayed in the left hand panel and the commenting panel on the right.

Scene Navigator

The screenshot displays the Scene Navigator interface. On the left, a script editor shows a scene titled "[1] INT. TEST LOCATION - DAY". The script includes instructions for using the scene editor, such as "USING THE SCENE EDITOR - THIS IS A CAPTION" and "A caption is indicated by enclosing the text using the 'pipe' symbol". It also shows character actions and dialogue, including a character named BOND (smiling) and a character named BAND. The script ends with "CUT TO:". On the right, the "Scene Comments" panel is active, showing a text input field and four comment types: Comment, Highlight, Question, and Idea. Below the input field, four comments by Stewart McKie are listed, each with a different icon and timestamp: a question mark for a question, a star for a highlight, a speech bubble for a comment, and a lightbulb for a suggestion.

Figure 6.15 – ScenePad4 Scene Commenting

This means that if a script/scene is shared with a group of users, say a screenwriting class, multiple users can easily comment on the same scene at the same time as part of a scene feedback and development assignment for example.

Screenplay Mobile Access and Delivery

Today's social networkers are used to interacting with their social network from mobile devices, especially their smartphones. They tweet from their phone they submit pictures taken with their phone camera, they rate or post

comments on content they view on their phone, from their phone.

Scenepad4 recognizes this need for mobile access to screenplay data and the potential utility of contributing data to a screenplay from a mobile device. This begins with generating a Quick Response code (QR code) for each script to enable content to be accessed from mobile devices (see figure 6.16). The script owner can then copy the QR code from Scenepad and paste it wherever it might be utilized from e.g. a blog or Facebook page. A smartphone user with a QR reader app installed can simply snap the code to access the screenplay content via the Scenepad mobile site.

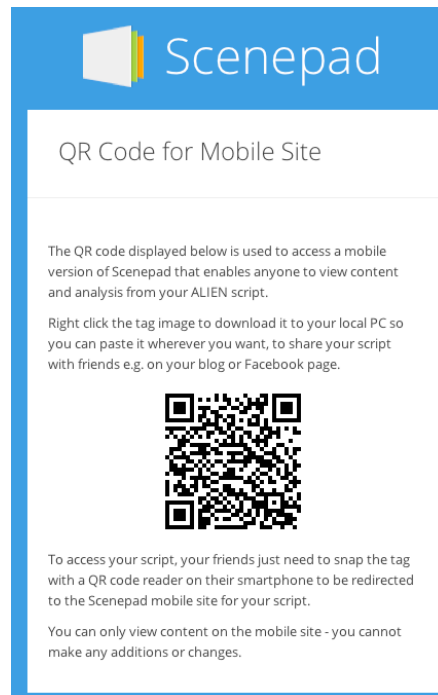


Figure 6.16 - Scenepad4 Script QR code

The Scenepad mobile site provides view only access to script content (scenes, roles, locations) and to a range of Scenepad analytics including charts and word clouds (see figure 6.17).

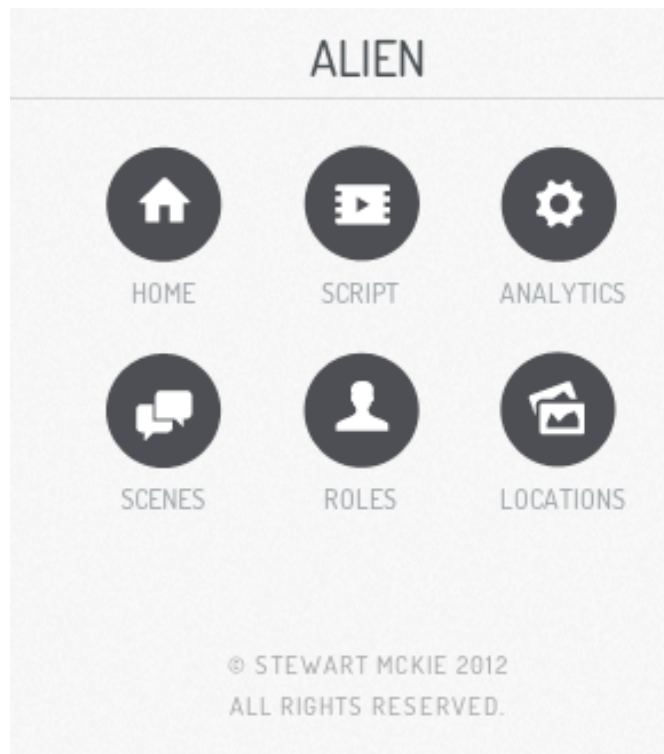


Figure 6.17 - Scenepad4 Mobile site home page

The Scenepad mobile app allows users to 'pull' content and analytics via their mobile device but does not yet allow users to push content directly into Scenepad via their mobiles devices, for example enabling content rating from the mobile app or pushing photos taken on the phone camera into a Scenepad storyboard. This is perfectly practical but would require a more sophisticated app on the device and ideally the addition of an application programming interface (API) to Scenepad.

One workaround for posting content to Scenepad is to allow Scenepad users to post content to other online applications that do have a more sophisticated smartphone app and an API and then 'pull' the data from these apps into Scenepad. Please note that as this functionality depends on 3rd party APIs that frequently change, the Scenepad functions may not continue to work as expected, or at all. There are three examples of API-based interaction with 3rd party sites in my Scenepad prototypes.

In Scenepad2 users could post any kind of data to a Tumblr blog, link it

somehow to specific Scenepad 'categories' (e.g. a role name) so that Scenepad could connect to their blog via the Tumblr API and display all the data for a specific category assembled onto a Scenepad Tumblr 'wall'. Figure 6.18 shows some images posted to the Scenepad Tumblr blog of potential actors for the casting of the role of *Maggie* in one of my own scripts. The images were posted to Tumblr and categorized as <scriptname>-<role>-<rolename> so that they can be retrieved from Tumblr and displayed in the correct Scenepad wall (in this case the wall for the role 'Maggie').

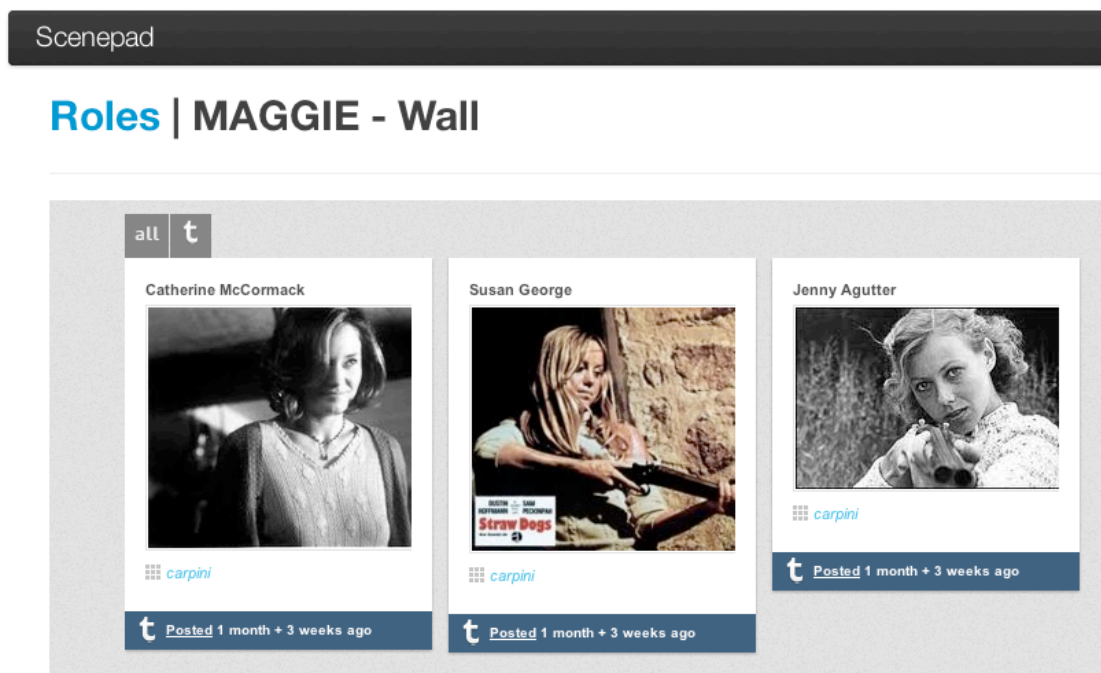


Figure 6.18 - Scenepad2 - showing Tumblr-sourced images displayed on Scenepad role wall

In Scenepad 3 & 4 users can display photos posted to a kee.ps account via Scenepad. Kee.ps is an online service like Flickr and others that lets you take a photo on your smartphone and post it to your online photo portfolio where they can be stored in different directories. So if you create a directory (e.g. 'Alien' on kee.ps to represent your script, then Scenepad can use the kee.ps API to retrieve these photos and display them as shown in figure 6.19.

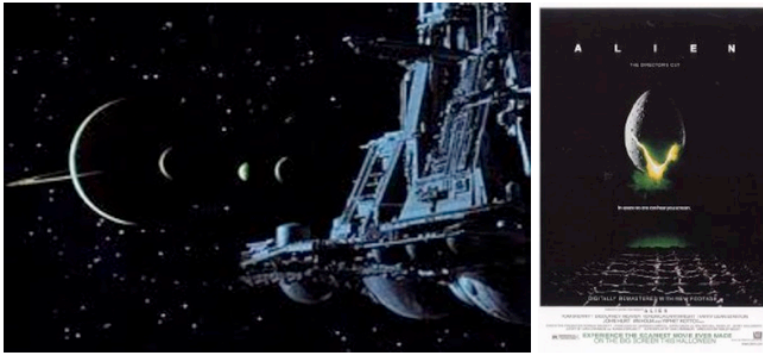
Keeps

Figure 6.19 - Scenepad4 - showing Images sourced from Kee.ps displayed under script Keeps tab

A third example of using an API to access 3rd party site content is the displaying of tweets linked to a script by use of the Twitter API. By setting up a user account on Twitter, say using the name of your script, multiple users can then post Tweets to this account and these tweets displayed in Scenepad as a timeline as shown in figure 6.20.

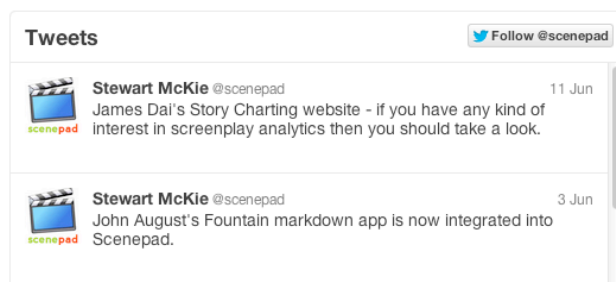
Tweets

Figure 6.20 - Scenepad4 - showing Tweets sourced from Twitter displayed under Script Tweets tab

Theoretically any content posted to a 3rd party site, with an API that Scenepad can use, could be retrieved and surfaced in Scenepad, significantly extending the possibilities of the screenplay as content hub.

6.4 Scenepad Analytics

The design of Scenepad is script data-centric rather than script-document centric. The content of the script is contained in the text of individual data snippets, which are in turn defined contextually by their snippet and scene metadata. The snippet text is essentially of two types: dialog and action. Other snippet types, such as 'caption' or 'transition' function as metadata that relate more to scene contextualization and the filmic flow of the content. It's worth remembering that an important purpose of content analytics is to give anyone about to read the script an impression of the script content without needing to read it first and to help writers 'see the forest from the trees'.

So in what ways and for what purposes can this Scenepad snippet text be analyzed?

Entity Grouping

Scenes are an obvious entity to group in order to get some insight into various aspects of the 'balance' of a script. For example, scenes can be grouped as internal/external and day/night. This helps give an immediate feel for script balance that may have genre implications e.g. more night than day scenes might be expected in a horror, or may indicate budgetary considerations e.g. lots of external night scenes may be more difficult and expensive to film. Scenes can also be grouped to reflect their respective dialog/action content. This helps to identify a potentially 'dialog-heavy' script and those scenes that stand out as either dialog or action-centric.

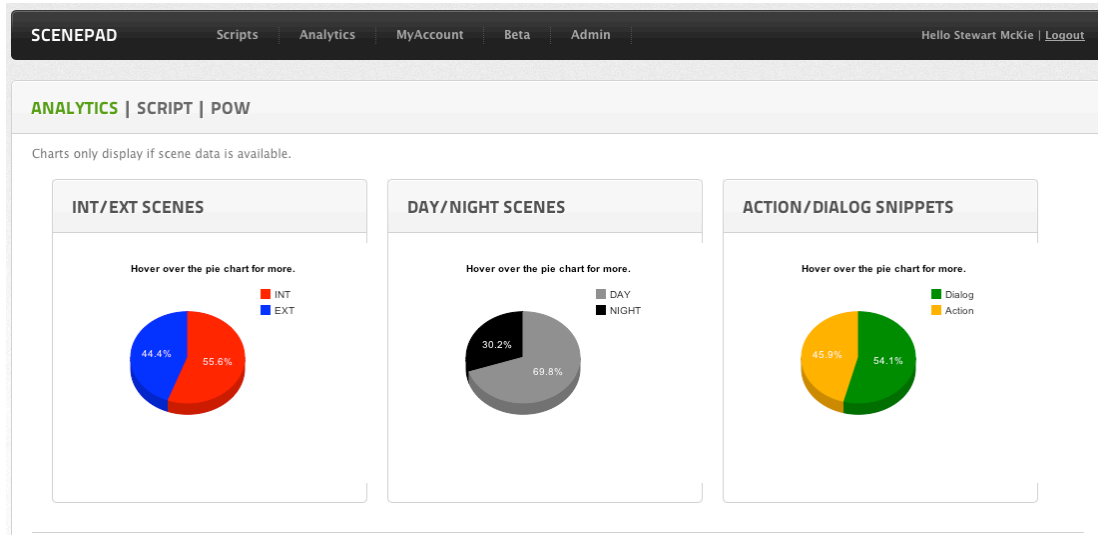


Figure 6.21 - Scenepad1 Script Dashboard

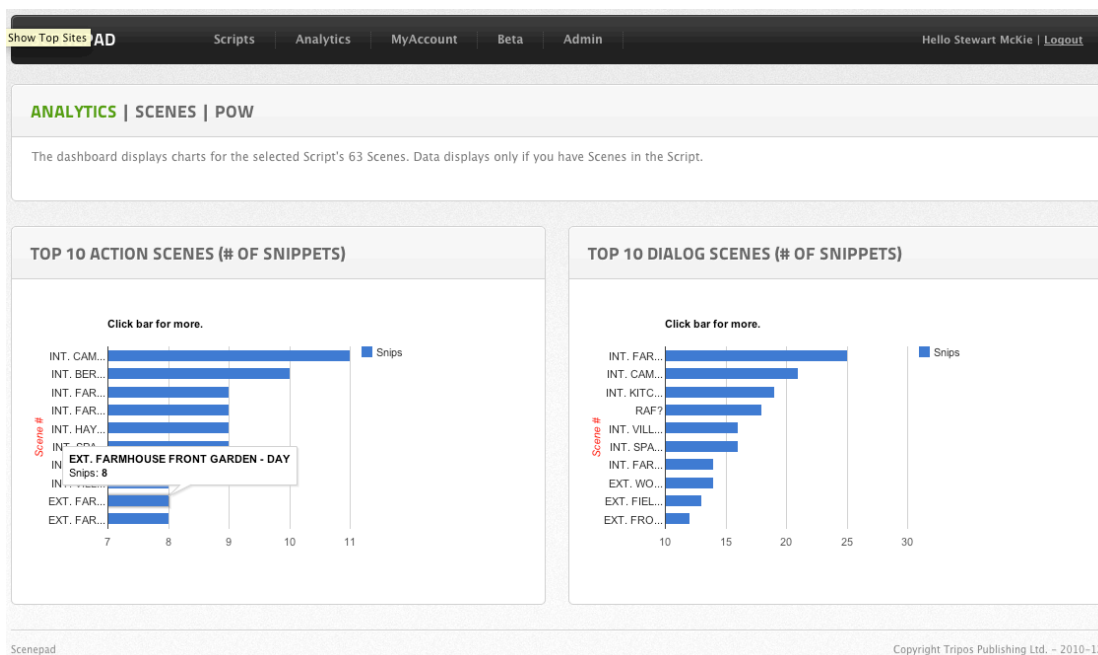


Figure 6.22 - Scenepad1 Scene Dashboard

In all screenwriting software, the scene content is a collection of snippets. Other ways to group snippets are by role or location. For example, in Scenepad you can view all dialog snippets by role (role-dialog) and all action snippets by location (location-action). The purpose of this simple grouping analysis is to help the writer(s), the talent and set/location managers focus on a specific POV of the script.

The writer is helped because role-dialog grouping facilitates establishing consistency of tone for a given role. A quick skim of a role-dialog list will help to identify any dialog that seems 'out of character'. For the talent playing the role, the role-dialog list isolates all their dialog across the whole script so it's easier to get a feel for their complete part, from a dialog perspective. The location-action grouping isolates action sequences within a specific location, action that may span multiple scenes using the same location, so that production managers can better plan their assets to realize the action.

ALIEN

👤 RIPLEY - Dialog Snippets

#	Text	Parenthesis	Voice	Scene	
21	How's it going.	voice over		INT. BLISTER - DAY	🔍 ✎
22	Have you tried putting the transmission through...			INT. BRIDGE	🔍 ✎
23	I'll give it a shot.			INT. BRIDGE	🔍 ✎
24	Ash, tell Dallas Mother speculates that the noise...			INT. BRIDGE	🔍 ✎
25	I'm going out after them.			INT. BRIDGE	🔍 ✎
26	I still think we should go after them.			INT. BRIDGE	🔍 ✎
27	How many.			INT. BRIDGE	🔍 ✎
28	Dallas, Lambert. Can you read me.			INT. BRIDGE	🔍 ✎
29	Okay.	voice over		INT. PASSAGEWAY NEAR AIR LOCK	🔍 ✎
30	Right here.			INT. BRIDGE	🔍 ✎
31	What happened to Kane. I need a clear definition.			INT. BRIDGE	🔍 ✎
32	If we let it in, the ship could be infected.			INT. BRIDGE	🔍 ✎

Figure 6.23 - Scenepad4 Role-Dialog

By grouping action/dialog snippets by role, writers and talent can get a quick idea of relative part size and by identifying a role as protagonist or antagonist writers can get an idea of the balance of these two key roles and when they enter and exit the script.

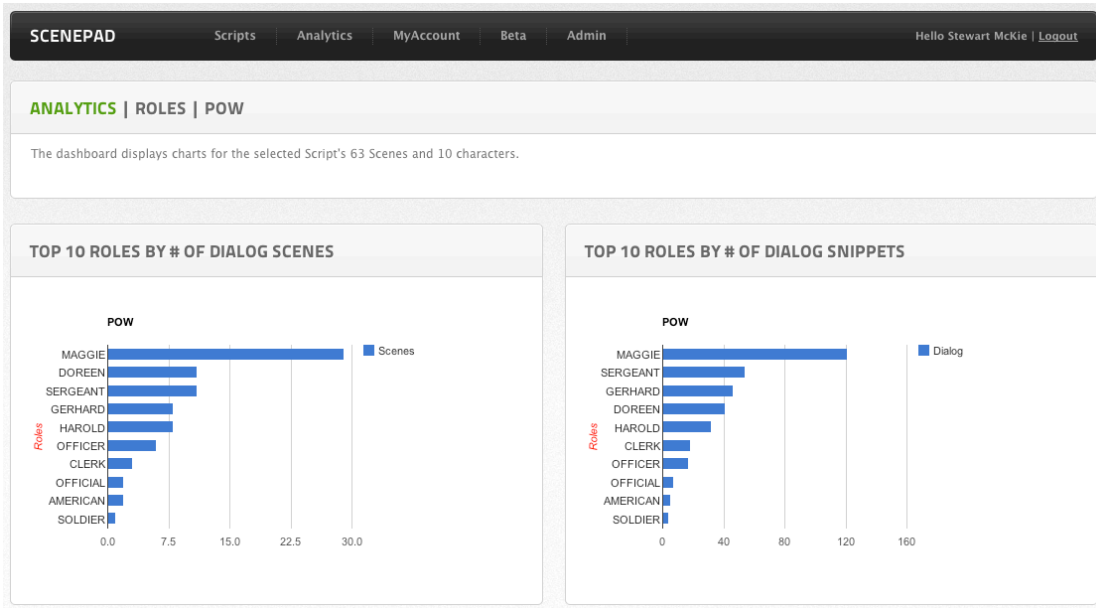


Figure 6.24 - Scenepad1 Top 10 Roles

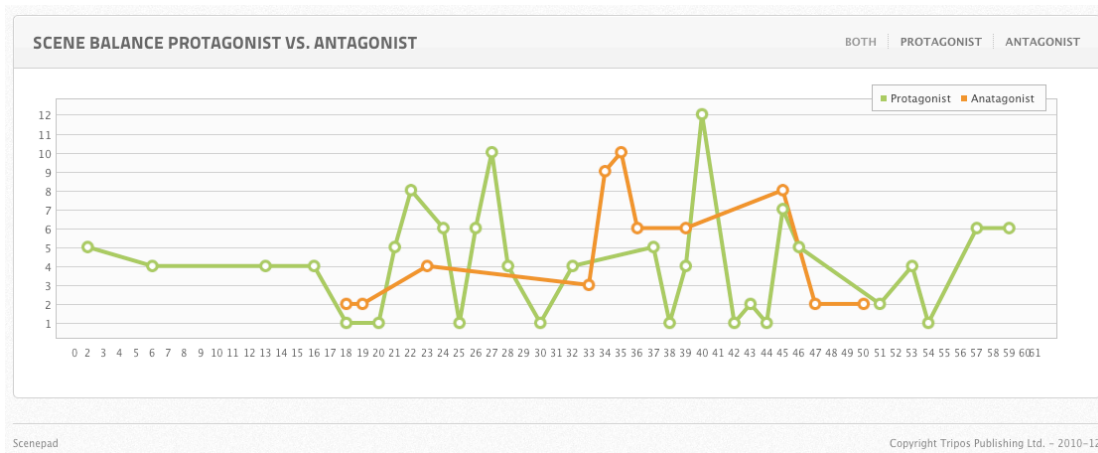


Figure 6.25 - Scenepad1 Protagonist vs. Antagonist

Locations can be grouped by their use in the script and by the number of roles that are involved at the location. This provides some insight into the most popular and 'crowded' locations in the script.

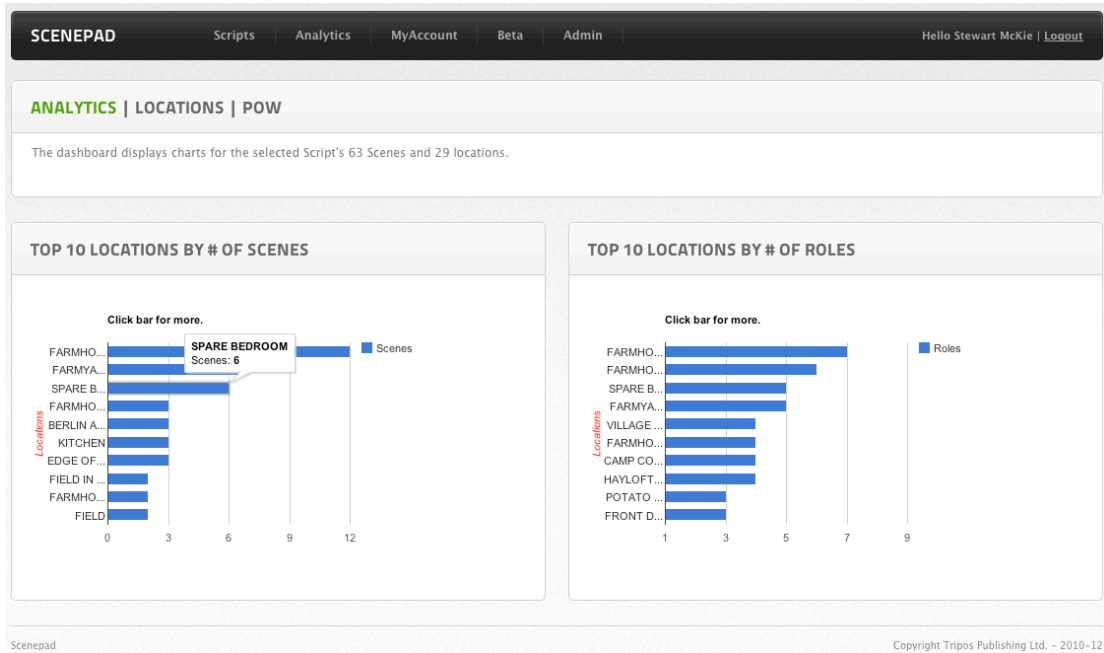


Figure 6.26 - Scenepad1 Top 10 Locations

Snippet Text Analysis

The grouping analytics discussed above is not focused on the actual text and individual words contained in action/dialog snippets, whereas snippet text analysis is. Scenepad provides a range of word clouds to help analyze textual content. The word clouds are generated at various levels:

Scriptcloud – word clouds of all text in the script

Dialog Cloud – a word cloud of all dialog text in the script

Action cloud – a word cloud of all action text in the script

Noun cloud – a word cloud containing only nouns

Verb cloud – a word cloud containing only verbs

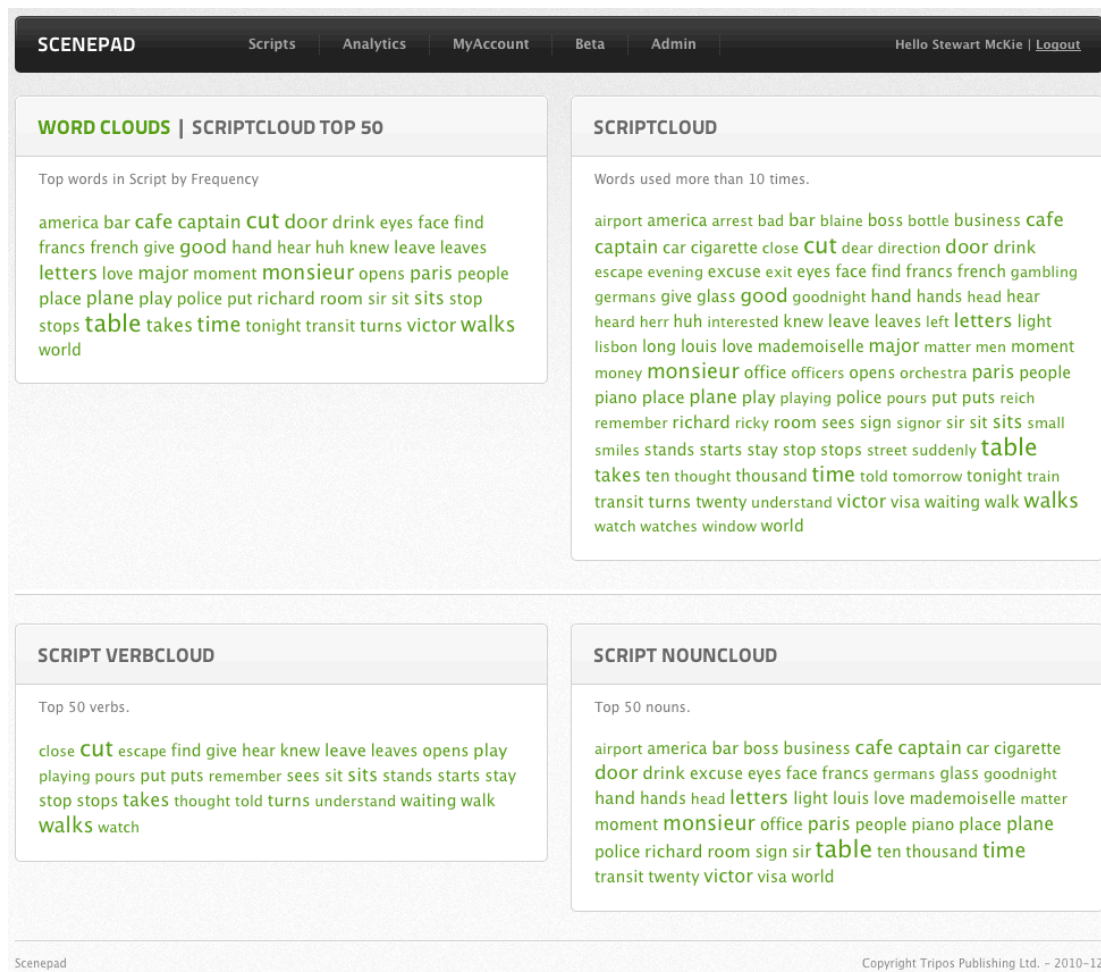


Figure 6.27 - Scenepad1 Script-Level Word Clouds

Like those on the original Scriptcloud site, all the Scenepad word clouds are subject to a 'stop-list'. And like Scriptcloud, the Scenepad scriptclouds give an 'impression' of the vocabulary of the script by displaying 'top n' words by frequency or those words that are used over a certain limit e.g. all verbs used more than 5 times.

Unlike Scriptcloud, Scenepad provides forms for the user to set the topN value (e.g. 20 or 50) and frequency limits and to manage (insert and delete) stoplist words. A relatively easy addition would be to enable a script-specific stoplist – for example to handle non-English scripts since the default stoplist includes 666 common English words only.

Unlike Scriptcloud, Scenepad provides some ability to extract both Noun and Verb clouds from the script textual content. This is done by using a lexicon of

tagged parts of speech based on the Brown Corpus, compiled in the 1960s by Henry Kucera and W. Nelson Francis at Brown University, Providence, Rhode Island. The code used to parse the source text and compare it to the lexicon is based on the tagger code originally written by Eric Brill and subsequently modified by Mark Watson.

Essentially words from the script content are compared to the part of speech classification of the lexicon word tags (e.g. VB to VBZ for verbs and NN to NR for nouns) and included in the Noun and Verb clouds as appropriate. If a word is tagged with both noun and verb tags then the primary tag is used to define which cloud the word appears in. For example, the lexicon entry **Aid NNP NN VB** is classified as both a noun and a verb but would be selected into the noun cloud rather than the verb cloud whereas **Aim VB NN NNP** would be selected into the verb cloud due to the order of part of speech identifiers. This simplistic approach will inevitably result in some words being misclassified so the word clouds are unlikely to be accurate every time.

The addition of verb and noun clouds helps to also give an impression of the vocabulary of action and dialog snippets from the perspective of two key parts of speech contained in a screenplay. Most nouns in a script can be directly mapped to a 'prop' used in a production set. Most verbs in a script are indicative of the kind of action that the script embodies.

A potential application of noun and verb analytics is for the purposes of a marketing firm analyzing a script for product placement purposes – to identify a script that offers a context that presents an opportunity for specific product placements. Sternberg (1997, p.49) gives a specific example of an analysis of the 2nd draft of the script of *Thelma and Louise* (1991) by product placement agency *Prime Time Marketing* that indicates how the availability of noun and verb clouds could have helped the agency make a decision about whether and where to use product placement in this script.

A potential application of verb analytics is to train an actor in a particular

kind of action so they can perform better or in a specific 'signature' way. A high proportion of verbs in a script is also likely to indicate an action-filled script, which in turn may help to confirm or deny the genre pretensions of the author.

Thrulines

Scenepad3 includes thrulines for storylines, keyroles (antagonist and protagonist) and nominated characters. The thruline simply visualizes scenes that have been tagged as belonging to a storyline or that specific roles or characters appear in.

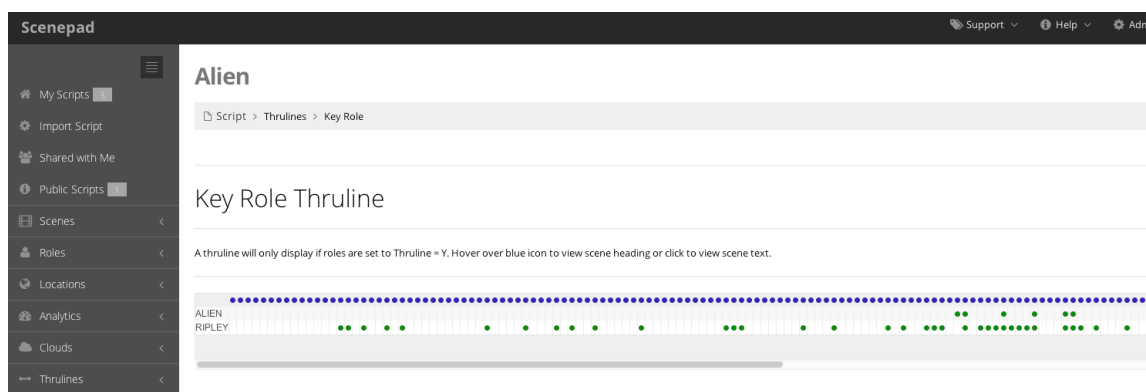


Figure 6.28 - Scenepad3 - Alien Keyrole Thruline

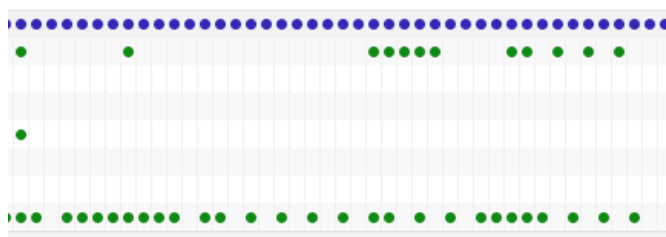


Figure 6.29 - Scenepad3 Alien End of Script Thruline

The set of thrulines for characters in *Alien* shows how, towards the end of the script, Ripley is the only crew member left, which either indicates total focus on a single character towards the end of the script or the situation in *Alien* where all other characters are dead. In figure 6.29 we see how in scene 222 (in this version of the script) the Alien disposes of Dallas (see dot in row

5) leaving just it (row 2) and Ripley (row8) remaining. Note that Ripley is the only dot left at the end of the script (the ship's cat not being a formal character).

Sentiment Analysis

Scenepad3 includes some sentiment analysis capability based on functionality provided by J W Hennessey's phpInsight class (see <https://github.com/JWHennessey/phpInsight> - accessed 18 June 2013). The sentiment analysis is either of dialog or action snippets within a script and provides a positive, neutral and negative rating for each textual snippet. The analysis is based on 'a dictionary of words that are categorised as positive, negative or neutral and a naive bayes algorithm to calculate sentiment. To improve accuracy phpInsight removes 'noise' words.'

As with the stop list applied to word clouds this dictionary largely determines the polarity of the text and so the efficacy of this dictionary is essential to the accuracy of the analysis.

Alien Sentiment Scores - Dialog

This page uses J W Hennessey's [phpInsight](#) class to deliver the sentiment score from the Script Dialog. Thank you James and Ismayil.

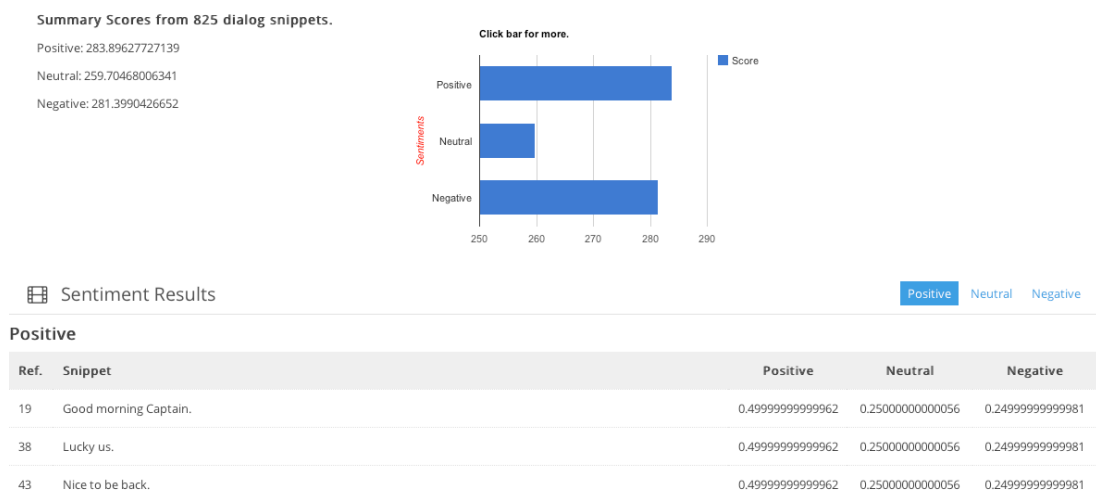


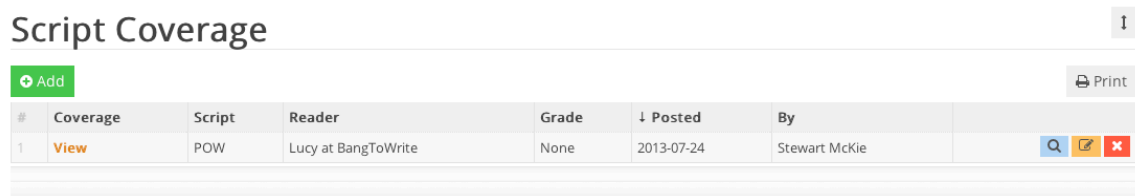
Figure 6.30 - Scenepad4 Dialog Sentiment Analysis

Figure 6.30 shows a sentiment analysis of every snippet in the Alien script

(i.e. its positive, neutral, negative score). This Scenepad sentiment analysis could be improved in future by applying it to the content of scenes in order to chart the positive/negative/neutral flow of the script scene-by-scene.

6.5 Scenepad for Business

Scenepad recognizes that screenwriting is a business process by providing functionality for tracking script coverage (see figure 6.31) and submissions; for posting time and billing to a script and for managing 'to-do' tasks related to a script. This enables much of the business of screenwriting to be managed from within Scenepad rather than in a separate application or spreadsheets. Again, as mentioned with storyboarding above, a 'coverage' user role could be created that only allows viewing of scene content in order to allow external readers to easily contribute coverage to the script.



#	Coverage	Script	Reader	Grade	↓ Posted	By	
1	View	POW	Lucy at BangToWrite	None	2013-07-24	Stewart McKie	<input type="text"/> <input type="button" value="🔍"/> <input type="button" value="✎"/> <input type="button" value="✖"/>

Figure 6.31 – Scenepad4 Managing Script Coverage

Scenepad also helps to manage the overall ownership of a script that may have been written and rewritten by multiple parties. Based on the ownership of a snippet, Scenepad provides a dashboard that shows how much of the script is owned by the top 5 users who have worked on the script – in terms of the whole script content and the action or dialog content only, as shown in figure 6.32.

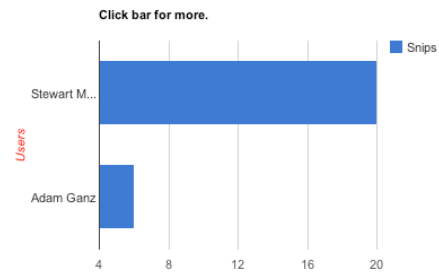
Snippet Ownership Analytics for Scenepad Demo/Help script

Charts only display if snippet data is available.

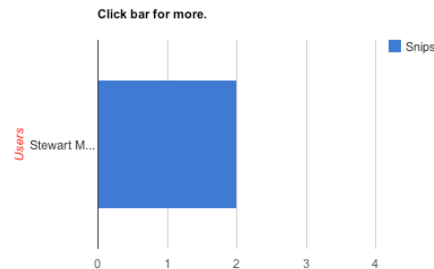
Script Snippet Stats

- 26 Script Snippets
- 2 Dialog Snippets
- 19 Action Snippets

Top 5 Owners All Script Snippets



Top 5 Owners by Dialog Snippets Only



Top 5 Owners by Action Snippets Only

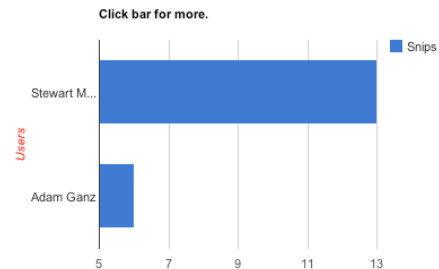


Figure 6.32 - Scenepad4 Script Ownership dashboard

6.6 Screenplay Production

Scenepad recognizes that screenwriting and screenplay realization is a long-duration, social process that includes the production of the movie. Most screenwriting applications do not. So they do not include the kind of production management capabilities found in packages such as *Gorilla* or *Chimpanzee*, both from Jungle Software.

Scenepad includes a complete production management sub module for managing shoot days, cast, crew and props, scene storyboards and production budgets for example. In this way Scenepad supports multiple social networks by using the datafied screenplay as the network content anchor. One social network is focused on the writing/rewriting of the screenplay content and another is focused on the realization of the screenplay content and an audio visual work.

By linking to and leveraging an online photo archiving site (kee.ps.com), the production module also allows practitioners to take on-set/on-location photos with their mobile phones, upload them to kee.ps and then access these photos through Scenepad as a record of the shoot. A 'shootstream' if you will.

RECONCILIATION | SHOOT DAY MANAGER Print Report

Tuesday: 23-July-2013

Location Information

We are filming at Shaftesbury School, during the school holidays.
 Catering - Provided

Contact: Fred Parley-Caretaker
 Phone: 01747-844566
 Shaftesbury School, Salisbury Street
 Shaftesbury SP7 9PL
[View Link](#) | [View Map](#)

Call Times

★ Cast Call Time - 06:00:00
 🏠 Crew Call Time - 08:00:00

Manage Set Photos on Keeps

Keep a record of your shoot on [Kee.ps](#) via their mobile apps or...
 ✉ Email today's on-set photos to cfoinfo.20130724@kee.ps

Shoot Day Content Activity | Addresses | Notes | Cast | Crew | Props | Keeps

Activity

Add Print

#	Day Date	Call Time	Type	Name	Scene	Status	Posted	By	
1	2013-07-23	08:00:00	Other	Scene Shoot	0 INT. BERLIN APARTMENT - DAY	Scheduled	2013-08-06	Stewart McKie	★ 🏠 🔍 📧 ✖
2	2013-07-23	12:00:00	Other	CGI IMAGING		Scheduled	2013-08-06	Stewart McKie	★ 🏠 🔍 📧 ✖

20

Figure 6.33 - Scenepad4 Script Day Content Hub

Summary

The various Scenepad prototypes show that using a 'datafied' script as a foundation enables many of the screenwriting 2.0 concepts discussed above to be applied and realized in screenwriting software. Scenepad4 provides significantly more screenplay analytics and visualization functions than are available in any of today's commercial screenwriting applications.

Scenepad4 delivers more social networking capabilities than any of today's screenwriting applications. Scenepad4 supports more screenplay structural capabilities that most of today's screenwriting applications.

Scenepad4 is a fully working prototype that shows the potential of the next

generation of commercial screenwriting 2.0 applications. It contains imported 'reference' scripts e.g. *Alien* and has been used, by me, to create a full-length screenplay script from scratch (*Reconciliation* – referred to in some of the figures above). Scenepad is only an initial step towards screenwriting 2.0 and I fully understand that many screenwriters would neither be interested in nor use the kind of functionality that Scenepad provides. But it does deliver some indication of 'what happens next' and why the datafication of screenplays is inevitable.

7. What Happens Next?

To return to Mark Norman's question cited at the beginning of this thesis, the datafication of screenplay content – the transition from a document-centric to a data-centric screenplay - has many interesting implications for screenwriting as a practice, as a research area and for the development of a new generation of screenwriting 2.0 software applications.

But first it will require moving on from the kind of thinking, referred to by Edgar (2009:6-7) in *How Plays Work*, that persists in the idea that creative works are not subject to rules: 'writers who've read any twentieth-century literary theory are understandably irked by the arithmetical reductionism of so much thinking in this field, with its mechanical lists, symbols, charts and graphs.' Edgar reflects that rules are 'a sedimentation of all of the plays (and, to an extent, all of the stories) that we have ever encountered' and, at some level, reflect what audiences expect to see. Here, Edgar is referring to play scripts and a theatre audience but the same principles apply equally well to the structure debate around screenplays and a cinema audience.

By moving beyond this kind of thinking, designers/developers of new screenwriting applications may reflect approaching the screenplay and screenwriting from different 'start points' than that of traditional screenplay application designers.

For example, it is much more likely that application designers will approach screenwriting from a social networking or analytics perspective when they come to design the next generation of screenwriting applications, rather than as simply a way to write and print out a 'correctly' formatted script. The influence of games designers may mean that screenplay applications are approached from the 'start point' of playing the script and managing the branching storylines of a user-controlled narrative for example. Or the pervasive use of mobile devices may mean that screenplays are 'reverse-engineered' from a series of images being linked together using simple apps

akin to Disney's *Story* (see <http://story.us>) where, in essence, the image is used as the stimulus to create the text rather than the other way around.

Something that screenplay datafication will also help to prevent is the screenplay as pre-formed artefact becoming anymore redundant than improvisational directors have made it already.

Back in 1974, Metz (1974:201-204, referring to Marcel Martin) discusses whether a "film-makers cinema" was replacing a "scriptwriter's cinema". Referencing the work of French director Jean Luc Godard, Metz suggests that Godard 'is one of those men for whom inspiration can only be fired during the shooting' and that it is during the shooting that the scenario is 'born' as a consequence of the film. In his section 'Ban the Script?', Lee (2013:6-10) refers to the improvisational preferences of British director Mike Leigh and Danish director Lars Van Trier's Dogme movement whose 'philosophy was initially to get totally away from the script.'

This might suggest that the future of the screenplay as an artefact is in doubt. However, this is unlikely to be the case if all screenplays are datafied. In fact, in this case, it is possible that it is filmmaking as solely human-directed process that is in doubt. Pre-viz software is the pathfinder here. If screenplays are datafied, and especially if they are datafied using a standardized format, then it is only a matter of time before ever-more sophisticated algorithms can interpret this data more effectively and start to generate 'first cut' movies (initially probably only animations), directly from the script. The job of the production team is then to take this first attempt and develop it either directly (in video) or indirectly first by making changes to the screenplay, thus changing the data, and re-generating the movie from the script. Somehow this does not feel like it is many years off.

In this scenario the screenplay is not something you can 'get way from' since the screenplay data is the movie or at least a significant part of it. CAD/CAM becomes SGF or screenplay generated filmmaking. And this will not just apply to animations, where the utility and practicality of this approach is

obvious. It's not hard to see that with the databases of a Google providing access to images from across the web, real-world 'streetviews' and satellite images of almost everywhere on earth or a YouTube full of previously released commercial movies and clips, and millions of user-generated 'home' movies and clips there is already a vast archive of digital content for a screenwriting application to search and retrieve from.

For example, take a single action snippet from a 'human' movie or a hybrid 'humanimation', combining human actors with CGI background, that can be translated into a short sequence or a single shot like:

They paused and watched the setting sun disappear into the Ocean.

or

There it is! A 1920 yellow Rolls-Royce.

The sunset clip or the clip of a Rolls-Royce could simply be retrieved from the online database and inserted into the movie's screenplay database as a link to the online clip that is in turn linked to the 'action' snippet data record.

Basically this only requires a search-and-retrieve algorithm that processes the screenplay data and proposes clips or images by 'reaching out' to online databases that the software automatically inserts into the proto movie's timeline. The 'filmmaker' can then choose from these 'suggestions' or use them as input to inform a more refined search of their own.

This promises a new kind of automated 'bricolage' to reference Wells' article in *Analysing the Screenplay* (2011) that takes the idea of Ruecker et al. of 'playing' the theatre play script (see http://academia.edu/3155085/WATCH_MY_MOVES_FROM_DIGITAL_PLAYS_TO_THE_DIGITAL_PLAYBOOK - accessed 11 July 2013) to a whole new level.

Aside from automated movie generation from a script, which is surely the

most important new development that screenplay as data can enable, the future of screenwriting technology and move to screenwriting 2.0 is likely to be characterized by a number of other innovations resulting from the improved application of technology to the screenplay as artefact or screenwriting as a process.

Innovation could emanate from the use of 'cloud' technology both to store and share scripts (already widely in use now by many online screenwriting applications like Scenepad) and application programming interfaces or APIs (surfaced as web services) for deploying and accessing applications, mashing them together and storing and accessing content; the pervasiveness of social networking, collaboration and crowdsourcing; and the form factor and new user interface (UI) controls - including touch, swipe and pinch - offered by mobile platforms such as the iPhone and iPad.

The new tactile UIs are not just changing the way we work with content on handheld devices. Large, wall-based, touch screens are also an interesting possibility for placing the script, and specifically scene content within a script, at the centre of a display of content linked to the script/scene. This kind of large tactile surface was glimpsed in *Minority Report* (2002) and more recently in *Den Som Draeber* (2010) and is implemented in the real world by Hornall Anderson and Nike among others. These 'touch-walls' present a content 'canvas' that could enable writing teams to work in a visual, collaborative and tactile way with both text and images to develop script content and re-fashion, rather than just rewrite, content relationships with the swipe of a hand. Dynamic, node-based representations of content and relationships, such as those facilitated by arbor.js (<http://www.arborjs.org>) are one dynamic way to present data on this kind of touch-wall.

Screenwriting 2.0 assumes that a screenplay is a dynamic, online and interactive audio-visual-textual artefact that is subject to continual enhancement and enrichment. It is not a paper-based document but a digital artefact subject to an ongoing development process that many roles, other

than 'writing' roles can participate in and contribute to as part of the screenplay development project team. Clearly, these stakeholders will be able to interact with this screenplay artefact in many different ways and on or from many different devices - something that will inevitably also impact the process and practice of screenwriting.

By framing screenwriting as a multi-user business - rather than only a single-user creative, process - future screenwriting applications must look and feel more like a social networking application rather than a document writing application. By recognizing that screenwriting is a process, writers may be encouraged to view their screenplay as a social artefact and expect that their content can benefit from exposure to different stakeholders in the their screenplay community of interest.

The screenplay itself will become an interactive online text where the anchoring reality of the scene text is augmented by layers of additional linked content that surrounds and enriches this text. Of course the printed script will not go away anytime soon but how long can it be before the digital script on an iPad essentially replaces the printed document on set and augmented reality screenplays become the norm?

Much of this linked content will come directly from user-generated content captured on mobile devices. These devices will act as the primary means to capture and connect content to the screenplay such as photos, audio and video clips. This content will flow-through to the screenplay via an application programming interface (API) that enables any external application to interface to it - both to put data into the screenplay and to get data from it. In this sense, the screenplay becomes the analog of the Facebook 'wall' that a community of interest engages with to enhance and enrich the core content.

This community engagement could prove especially useful in the production phase of the screenplay realization process for activities such as location management, shot planning and script continuity. The mere fact that it will

become easier to capture and connect this content will itself mean that the augmented screenplay will inevitably become a reality in the same way that a Twitter timeline or Flickr portfolio receives a constant stream of fresh content from a user generation that is used to capturing all kinds of content whenever, wherever.

There's no reason why this new kind of datafied screenplay could not also be managed on its own dedicated device – a 'Screenplay Kindle' if you will. By tying the hardware to a known, datafied content format the device can provide additional capabilities that enhance the user experience when you interact with the content, for example:

- VCR Buttons to jump through the script by scene or by role or by location
- A button to trigger a Siri-like voice to read out the dialog within a scene
- A button to mark a note entry in the text and type in the note content

The ability to build a device dedicated to screenplay content would be a lot easier if a standard screenplay data format (e.g. an XML schema) was agreed that is 'open' and not 'owned' by a specific software vendor (e.g. Final Draft). Once a properly documented standard is adopted, this is likely to encourage the wider software development community to engage with screenplay content management as a domain and help to drive the development of low-cost add-on apps to further enhance the screenwriting experience.

Datafication of screenplays means that stakeholders will come to expect a great deal more from their screenwriting application in the way of analytics. Writers will expect more analysis to help them improve the quality of their script, financial stakeholders will expect more analysis to help them judge the commercial potential of a script, and script readers and analysts will expect more data-based evidential analytics to support or challenge their intuition-based judgments of a screenplay. Some of the visualizations that the how-to manuals have presented as diagrams will be able to be generated directly from the screenplay content at the click of a button.

Some of these visualizations should become more accurate and reliable over time. For example as further research determines the optimum stop list for a text cloud or the optimum positive/neutral/negative indicators for sentiment analysis not just for screenplays generally but specifically by screenplay genre. Indeed, a set of consistently datafied scripts within a given genre should provide a sufficiently interesting 'big data' corpus to drive a whole new research area - screenplay genre analytics - that could have significant impact on the definition of screenplay genres and therefore what constitutes genre conformity or subversion in a given screenplay instance. Once this online genre database is available individual scripts can be submitted for analysis and comparison to genre norms and some kind of 'genre conformance' report produced as feedback for the script writer/owner.

A library of screenplay algorithms could be developed in order to define common analytical tasks such as how to generate a role throughline or identify a turning point. The potential of event identification in a screenplay also seems a rich area for further research. If events can be recognized and subsequent patterns visualized there is real potential for the deep structure of a screenplay to be surfaced evidentially rather than intuitively. This could re-energize the screenplay structure debate in that this kind of data-driven emergent structure is potentially much more interesting to the screenwriter as questioning tool than attempting to write to some deliberately imposed n-act structure.

In terms of screenwriting 2.0, it is the datafication of screenplays that will fundamentally drive 'what happens next' and there is every reason to be excited about what that 'next' could be in terms of the screenplay as artefact and screenwriting as practice. My Scenepad application deliverable merely scratches the surface of all this potential but does deliver an indication of how the datafication of the screenplay can, I believe, positively impact creating and managing, presenting and sharing, analyzing and visualizing textual screenplay content.

Bibliography

- Anderson, Chris (2012). *Makers: The New Industrial Revolution*. London: Random House Business Books.
- Argentini, Paul (1998). *Elements of Style for Screenwriters*. Los Angeles: Lone Eagle Publications.
- Arijon, Daniel (1976). *Grammar of the Film Language*. New York: Focal Press.
- Armer, Alan A. (1993). *Writing the Screenplay*. 2nd Ed. Belmont, CA: Wadsworth Pub. Co.
- Aronson, Linda (2001). *Screenwriting Updated New (and Conventional) Ways of Writing for the Screen*. Los Angeles: Silman-James Press.
- Aronson, Linda (2010). *The 21st Century Screenplay*. Los Angeles: Silman-James Press.
- Aristotle, *Poetics*, trans. Malcolm Heath, London: Penguin, 1996.
- Baboulene, David (2010). *The Story Book*. DreamEngine Media.
- Batty, Craig and Waldeback, Zara (2008), *Writing for the Screen; Creative and Critical Approaches*. Basingstoke: Palgrave Macmillan.
- Borst, Terry (2012). *Mastering Celtx*. Boston: Course Technology.
- Bruhn Jensen, Klaus (Ed.)(2002). *A Handbook of Media and Communication Research: Qualitative and Quantitative Methodologies*. London: Routledge.
- Card, Stuart, MacKinlay Jock and Shneiderman, Ben (1999). *Readings in Information Visualization: Using Vision to Think*. London: Academic Press.
- Chatman, Seymour (1980). *Story and Discourse: Narrative Structure in Fiction*

and Film. Ithaca, New York: Cornell University Press.

Cockburn, Alistair (2005). *Crystal Clear: A Human-Powered Methodology for Small Teams*. Upper Saddle River NJ: Pearson Education Inc.

Cole, Hillis R., and Haag, Judith H. (1989). *The Complete Guide to Standard Script Formats – Part1: The Screenplay*. Los Angeles: CMC Publishing.

Cooper, Dona (1994). *Writing Great Screenplays for Film and TV*. New York: Prentice Hall.

Corrigan, Timothy J. (2007). *A Short Guide to Writing About Film*. New York: Pearson Longman.

Cowgill, Linda J. (1999). *Secrets of Screenplay Structure How to Recognize and Emulate the Structural Frameworks of Great Films*. Los Angeles: Lone Eagle Publications.

Cowgill, Linda J. (1997). *Writing Short Films Structure and Content for Screenwriters*. Los Angeles: Lone Eagle Publications.

Cron, Lisa (2012). *Wired for Story*. New York: Ten Speed Press.

Decker, Dan (1998). *Anatomy of a Screenplay*. Chicago: The Screenwriters Group.

De Haan, Erik (2004). *The Consulting Process as Drama. Learning from King Lear*. London: Karnac.

Donovan, John J., 'Three-tiered Client-Server Systems Gain as Enterprise-Wide Solution', Interview, *Investor's Business Daily*, (October 1994).

Edgar, David (2009). *How Plays Work*. 2nd Ed. London: Nick Hern Books.

Edwards, Rona and Skerbelis, Monika (2009). *I Liked It. Didn't Love It*. 2nd Ed. Beverly Hills: Edwards Skerbelis Entertainment.

Egri, Lajos (2004). *The Art of Dramatic Writing*. New York: Touchstone (original edition, 1942).

Eliashberg Jehoshua, Hui Sam K., Zhang John Z. From Story Line to Box Office: A New Approach for Green-Lighting Movie Scripts in *Management Science* Vol. 53, No. 6, June 2007, pp.881-893.

Elberse, Anita (2013). *Blockbusters: Why Big Hits – and Big Risks – are the Future of the Entertainment Business*. New York: Henry Holt

Epstein, Alex (2002). *Crafty Screenwriting: Writing Movies That Get Made*. New York: Henry Holt.

Epstein, Edward Jay (2012). *The Hollywood Economist: The Hidden Financial Reality Behind the Movies*. New York: Melvyn House Publishing.

Few, Stephen (2006). *Information Dashboard Design*. Sebastopol, CA: O'Reilly Media.

Field, Syd (2005). *Screenplay: The Foundations of Screenwriting*. Rev. Ed. New York: Delta Trade Paperbacks.

Flinn, Denny Martin (1999). *How Not to Write a Screenplay*. New York: Watson-Guptill Publications.

Forster, E. M. (1927). *Aspects of the Novel*. London: Penguin Books, 2005.

Franzosi, Roberto (2010). *Quantitative Narrative Analysis*. London: SAGE Publications.

Frensham, Ray (2003). *Teach Yourself Screenwriting*. 2nd Ed. London: Hodder Headline Ltd.

Freytag, Gustav (1899). *Technique of the Drama: An Exposition of Dramatic Composition and Art*. (trans. Elias J. McEwan). Amsterdam: Fredonia Books,

2005.

Ganz, Adam (2011). 'Let the audience add up two plus two. They'll love you forever' in Nelmes, Jill (ed.) *Analysing the Screenplay*. Oxford: Routledge, 127-141.

Garfinkel, Asher (2007). *Screenplay Story Analysis: The Art and Business*. New York: Allworth Press.

Gaurav, Vaish (2013). *Getting Started with NoSQL*. Birmingham: Packt Publishing.

Geroimenko, Vladimir, and Chaomei Chen (2006). *Visualizing the Semantic Web: XML-Based Internet and Information Visualization*. 2nd Ed. London: Springer.

Goldman, William (1996). *Adventures in the Screen Trade*. London: Abacus.

Gotham Writers Workshop Faculty (2006). *Writing Movies*. Ed. Alexander Steele. New York: Bloomsbury.

Grove, Elliot (2001). *Raindance Writer's Lab Write + Sell the Hot Screenplay*. Boston: Focal Press.

Hoad, Phil (2007). *Stars? Check. Script? Don't need one yet...* The Guardian (Film Section) 13 July 2007.

Horton, Andrew (1999). *Writing the Character-Centered Screenplay*. 2nd Ed. Berkeley: University of California Press.

Indick, William (2004). *Psychology for Screenwriters: Building Conflict in Your Script*. Studio City, CA: Michael Wiese Productions.

James, Henry (1884). 'The art of fiction' in Shapira, Morris (ed.), *Henry James: Selected Literary Criticism*. London: Penguin, 1963.

Kaplan, R.S. and Norton D.P. (1996). *Using the balance scorecard as a strategic*

- management system* in Harvard Business Review, January/February, 1996.
- Karetnikova, Inga (1990). *How Scripts are Made*. Carbondale IL: Southern Illinois University Press.
- Kawin, Bruce F (1992). *How Movies Work*. Berkeley and Los Angeles: University of California Press.
- Kenning, Jennifer (2006). *Writing a Great Movie: Key Tools for Successful Screenwriting*. New York: Continuum.
- Kitchen, Jeff (2006). *How to Be Your Own Script Doctor*. New York: Lone Eagle Publications.
- Lacey, Nick (2000). *Narrative and Genre Key Concepts in Media Studies*. Basingstoke: Palgrave.
- Langford, Barry (2011). 'Beyond McKee' in Nelmes, Jill (ed.) *Analysing the Screenplay*. Oxford: Routledge, 251-262.
- Lavandier, Yves (2004). *Writing Drama*. Trans. Bernard Besserglik. Paris: Le Clown & L'Enfant.
- Lee, Jason (2013). *The Psychology of Screenwriting: Theory and Practice*. London: Bloomsbury Publishing plc.
- Lewin, K (1946). Action research and minority problems. *Journal of Social Issues*, 2(4), 34-46.
- Liu, Zhi-Qiang. *Script Visualization (ScriptViz): a Smart System*. Dissertation. Centre for Lucey, Paul (1996). *Story Sense: Writing Story and Script for Feature Films and Television*. New York: McGraw-Hill.
- Mackendrick, Alexander, and Paul Cronin (2004). *On Film-Making: An Introduction to the Craft of the Director*. London: Faber and Faber.

MacDonald, Ian W. (2004), 'Disentangling the Screen Idea' in *Journal of Media Practice*, 5:2, 89–99.

(2010) '...So it's not surprising I'm neurotic' The Screenwriter and the Screen Idea Work Group' in *Journal of Screenwriting*, 1:1, 45-58.

Mamet, David (2007). *Bambi vs. Godzilla*. London: Simon & Schuster UK Ltd.

Maras, Steven (2009). *Screenwriting: History, Theory and Practice*. London: Wallflower Press.

Matteo, Civaschi and Milesi, Gianmarco (2012). *Life in Five Seconds: Over 200 Stories for Those With No Time to Waste*. London: Quercus Publishing Ltd.

Mayer-Schönberger, Viktor and Cukier, Kenneth (2013). *Big Data*. London: John Murray (Publishers).

McGinn, Colin (2005). *The Power of Movies How Screen and Mind Interact*. New York: Pantheon Books.

McKee, Robert (1997). *Story Substance, Structure, Style and the Principles of Screenwriting*. New York: Regan Books.

McKie, Stewart (2006a). 'Mariner Montage: Screenwriting software with a marketing edge' in *ScriptWriter*, September 2006, 49-51.

McKie, Stewart (2006b). 'Sophocles' in *ScriptWriter*, November 2006, 36-39.

Metz, Christian (1974). *Film Language: A Semiotics of the Cinema*. Chicago: University of Chicago Press.

Millard, Kathryn (2010). 'After the typewriter: the screenplay in a digital era' in *Journal of Screenwriting*, 1:1, 11-25.

Millard, Kathryn (2011). 'The screenplay as prototype' in Nelmes, Jill (Ed.) *Analysing the Screenplay*. Oxford: Routledge, 142-157.

Murdock, Maureen (1990). *The Heroine's Journey*. Boston: Shambala Publications.

Norman, Marc (2008). *What Happens Next. A History of American Screenwriting*. London: Aurum Press Ltd.

O'Bannon, Dan with Lohr, Matt R (2013). *Dan O'Bannon's Guide to Screenplay Structure*. Studio City, CA: Michael Wiese Productions.

Palmer, Frederick (1919). *Palmer Plan Handbook*. Los Angeles: Palmer Photoplay Corporation.

Parker, Philip (1999). *The Art and Science of Screenwriting*. Exeter: Intellect.

Pepperman, Richard D (2005). *Setting Up Your Scenes: The Inner Workings of Great Films*. Studio City, CA: Michael Wiese Productions.

Pfister, Manfred (1977). *The Theory and Analysis of Drama*. Cambridge: Cambridge University Press, 1991.

Pope, Thomas (1998). *Good Scripts, Bad Scripts Learning the Craft of Screenwriting Through the 25 Best and Worst Films in History*. New York: Three Rivers Press.

Price, Steven (2010). *The Screenplay: Authorship, Theory and Criticism*. Basingstoke, Hants: Palgrave Macmillan.

Propp, V (1968). *Morphology of the Folktale*. 2nd Ed. Austin, TX: University of Texas Press.

Rimmon-Kenan, Shlomith (2002). *Narrative Fiction*. 2nd Ed. London: Routledge.

Savin-Baden, Maggi and Howell Major, Claire (2013). *Qualitative Research: The essential guide to theory and practice*. Abingdon: Routledge.

Scofield, Sandra (2007). *The Scene Book: A Primer for the Fiction Writer*.

Penguin: London.

Seger, Linda (2003). *Advanced Screenwriting Raising Your Script to the Academy Award Level*. Los Angeles: Silman-James Press.

Sher, Lucy (2011). *Reading Screenplays: How to analyze and evaluate film scripts*. Harpenden: Kamera Books.

Siegel, Eric (2013). *Predictive Analytics: The Power to Predict Who Will Click, Buy, Lie, or Die*. New Jersey: John Wiley.

Silver-Lasky, Pat (2004). *Screenwriting for the 21st Century*. London: Batsford.

Sims, Chris and Johnson, Hillary Louise (2012). *Scrum: A Breathtakingly brief and agile introduction*. Dymaxicon.

Snyder, Blake (2005). *Save the Cat! The Last Book on Screenwriting You'll Ever Need*. Studio City, CA: Michael Wiese Productions.

Sternberg, Claudia (1997). *Written for the Screen: The American Motion-Picture Screenplay as Text*. Tübingen: Stauffenburg Verlag.

Suppa, Ron (2005). *Real Screenwriting Strategies and Stories From the Trenches*. Boston: Thomson Course Technology PTR.

Takach, Geo (2006). 'Projecting Knowledge Management: Screenwriting As A Tool For Knowledge-Sharing And Action' in *Journal of Knowledge Management Practice*, Vol. 7, No. 1, March 2006.

Tierno, Michael (2002). *Aristotle's Poetics for Screenwriters Storytelling Secrets From the Greatest Mind in Western Civilization*. 1st Ed. New York: Hyperion.

Tufte, Edward R (1990). *Envisioning Information*. Cheshire, Conn.: Graphics Press.

Vaish, Guarav (2013). *Getting Started with NoSQL*. Birmingham: Packt

Publishing.

Vale, Eugene (1944). *The Technique of Screenplay Writing*. London: Souvenir Press, 1973.

Van Sijll, Jennifer (2005). *Cinematic Storytelling the 100 Most Powerful Film Conventions Every Filmmaker Must Know*. Studio City, CA: Michael Wiese Productions.

Vogler, Christopher (1998). *The Writer's Journey: Mythic Structure for Writers*. 2nd Ed. Studio City, CA: Michael Wiese Productions.

Voytilla, Stuart (1999). *Myth and the Movies*. Studio City, CA: Michael Wiese Productions.

Wells, Paul (2011). 'Boards, beats, binaries and bricolage', in Nelmes, Jill (ed.) *Analysing the Screenplay*. Oxford: Routledge, 89-105.

Werbach, Kevin and Hunter, Dan (2012). *For the Win: How Game Thinking Can Revolutionize Your Business*. Philadelphia: Wharton Digital Press.

Wright, Kate (2004). *Screenwriting is Storytelling: Creating an a-List Screenplay That Sells*. New York: A Perigee Book.

Yanno, Drew (2006). *The 3rd Act: Writing a Great Ending to Your Screenplay*. London: Continuum.

Yorke, John (2013). *Into the Woods: A Five-Act Journey into Story*. London: Pearson.

Screenwriting and Other Software References

Amazon Studios.

<http://studios.amazon.com/storyteller> - accessed 11 June 2013.

BigML

<https://bigml.com> - accessed 18 March 2014.

Celtx

<http://www.celtx.com/> - accessed 23 January 2012.

Character Writer

<http://typingchimp.com/characterwriter/index.html> - accessed 23 January 2012.

Dataseed

<http://getdataseed.com> - accessed 18 March 2013.

Dramatica Pro

<http://www.dramatica.com/theory/articles/index.htm> - accessed 23 January 2012.

easySCOTT

<http://www.easyscott.de/en/easyscott.html> - accessed 17 June 2013.

Final Draft

<http://www.finaldraft.com/products/> - accessed 23 January 2012.

Final Draft AV

<http://www.finaldraft.com/company/about-us.php> - accessed 23 January 2012.

FrameForge

<http://www.frameforge3d.com/kb/entry/25/> - accessed 23 January 2012.

Google Sketchup

<http://www.sketchup.com> - accessed 01 February 2013.

Gorilla

<http://www.junglesoftware.com/home/> - accessed 23 January 2012.

Highland/Fountain

<http://johnaugust.com/2013/highland-ships> - accessed 29 March 2013.

Logline

<https://loglineapp.com> - accessed 29 March 2013.

Montage

<https://www.marinersoftware.com/discussions/forum/montage/> - accessed 17 June 2013.

Movie Outline

<http://www.movieoutline.com/> - accessed 23 January 2012.

PC-ACE

<http://sociology.emory.edu/faculty/rfranzosi/pc-ace/> - accessed 30 December 2013.

Persona

<https://www.marinersoftware.com/products/persona/> - accessed 29 March 2013.

Plotbuilder

<http://www.plotbuilder.com> - accessed 17 June 2013.

Script Supervisor

<http://peterskarratt.com> - accessed 17 June 2013.

ShotList

<http://solubleapps.com/shotlist/> - accessed 18 March 2014.

Sophocles

<http://en.wikipedia.org/wiki/Sophocles> - accessed software - accessed 29 March 2013.

Splunk

<http://splunk.com> - accessed 18 March 2014.

Storyboard Artist Studio

http://www.powerproduction.com/artist_studio.html - accessed 23 January 2012.

StoryO

http://www.junglesoftware.com/storyo/what_is_storyo.php - accessed 23 January 2012.

StoryView

<http://www.write-bros.com/> - accessed 23 January 2012.

Trelby

<http://www.trelby.org> - accessed March 29 2013.

Web References

85th Academy Awards.

<http://www.brandwatch.com/2013/02/introducing-the-social-oscars-tracking-award-buzz/>

<http://www.brandwatch.com/2013/02/the-power-of-crowds-predicting-the-oscar-winners/> - accessed 25 February 2013.

Agile manifesto. <http://agilemanifesto.org/principles.html> - accessed 05 February 2014.

Alexander, Christopher.

<http://www.patternlanguage.com/leveltwo/caframe.htm?/leveltwo/./bios/designpatterns.htm> - accessed 15 January 15.

Aman, Siama and Szpakowicz Stan. *Identifying Expressions of Emotion in Text* in V. Matousek and P. Mautner (Eds.): TSD 2007, LNAI 4629, pp. 196–205, 2007.

<http://www-scf.usc.edu/~saman/pubs/2007-TSD-paper.pdf> - accessed 12 May 2014.

Agarwal A., Balasubramanian S., Zheng J., Dash S. *Parsing Screenplays for Extracting Social Networks from Movies*. Proceedings of the 3rd Workshop on Computational Linguistics for Literature (CLfL) @ EACL 2014, pages 50–58, Gothenburg, Sweden, April 27, 2014.

<http://aclweb.org/anthology//W/W14/W14-0907.pdf> - accessed 12 May 2014.

Ayzenshtat, Mark, Elena Filatova, Kristina Holst, and Vladislav Shchogolev.

EMuse: Rapid Screenplay Prototyping Language. Dissertation. University of Columbia.

<http://www1.cs.columbia.edu/~sedwards/classes/2003/w4115/eMuse.final.pdf>

- accessed 07 February 2007.

Barber, Ian (20 November 2009). *Part of Speech Tagging*.

<http://phpir.com/part-of-speech-tagging/> - accessed 02 April 2012.

Beaudouin-Lafon and Mackay.

<https://www.lri.fr/~mackay/pdffiles/Prototype.chapter.pdf> - accessed 11 March 2014.

Bio printing

<http://www.explainingthefuture.com/bioprinting.html> - accessed 03 January 2013.

Blackstock A. and Spitz M. *Classifying Movie Scripts by Genre with a MEMM Using NLP-Based Features.*

<http://nlp.stanford.edu/courses/cs224n/2008/reports/06.pdf> - accessed 12 May 2014.

Boese, Steve. <http://steveboese.squarespace.com/storage/sample-screenplay-page.gif> - accessed 7 June 2013.

Box Office Mojo (Avatar). (<http://boxofficemojo.com/news/?id=2667> - accessed 30 August 2012.

boyd, d. m., and Ellison, N. B. (2007). 'Social network sites: Definition, history, and scholarship' in *Journal of Computer-Mediated Communication*, 13(1), article 11. <http://jcmc.indiana.edu/vol13/issue1/boyd.ellison.html> - accessed 09 July 2013.

Brill, Eric (1992). *A Simple Rule-Based Part of Speech Tagger.*

<http://acl.ldc.upenn.edu/H/H92/H92-1022.pdf> - accessed 02 April 2012.

Brown Corpus.

http://en.wikipedia.org/wiki/Brown_Corpus#Part-of-speech_tags_used - accessed 02 April 2012.

Chun, Russell (Story Visualizer).

<http://www.russellchun.com/storystructure/storyvisualizer.html> - accessed 15 January 2013.

Clark, Jeff. www.neoformix.com - accessed 11 June 2013.
<http://www.neoformix.com/Projects/TranscriptAnalyzer/index.html> - accessed 06 July 2008.

Clayton, Sue (2006). *Development Lab-The Importance of Good Peer Review in the Context of Screenwriting*.
<http://www.adm.heacademy.ac.uk/resources/case-studies/development-lab-the-importance-of-good-peer-review-practice-in-the-context-of-screenwriting/> - accessed 10 December 2012.

Coats, Emma. <http://slacktory.com/2012/07/pixar-story-rules-illustrated-by-icanlegothat/> - accessed 03 September 2012.

Couchbase. <http://www.couchbase.com/docs/couchbase-devguide-2.0/modeling-documents.html> - accessed 10 June 2013.

Creative Skillset. <http://www.skillset.org/film/jobs/script/> - accessed 04 September 2012.

Dai, James (Story Charting). <http://www.storycharts.ca/pages/casablanca/> - accessed 11 June 2013.

Davenport, G, and Murtaugh, M (1997). 'Automatist Storyteller Systems and the Shifting Sands of Story' in *IBM Systems Journal* 36 (1997): 446-456.
<http://www.research.ibm.com/journal/sj/363/davenport.html> - accessed 05 February 2007.

DiNucci, Darcy (1999). 'Fragmented Future' in *Print Magazine*. (April 1999): 32, 221-22. http://darcy.d.com/fragmented_future.pdf - accessed 15 December 2012.

Disney's Story. <http://story.us> - accessed 12 February 2014.

Donath, Judith S. (1995). 'The Illustrated Conversation' in *Multimedia Tools and Applications* 1

<http://smg.media.mit.edu/papers/Donath/IllustratedConversation.pdf> -
accessed 07 February 2007.

Eliashberg, Josh. *'Revenge of the Nerds,' Part V: Can Computer Models Help Select Better Movie Scripts?*. Knowledge@Wharton, November 29 2006.
<http://knowledge.wharton.upenn.edu/article/revenge-of-the-nerds-part-v-can-computer-models-help-select-better-movie-scripts/> - accessed 17 December 2006.

Eliashberg Jehoshua, Hui Sam K., Zhang John Z. *Green-lighting Movie Scripts: Revenue Forecasting and Risk Management*.
<http://forum.johnson.cornell.edu/workshop/MARKETING/Hui.pdf> - accessed 18 March 2014.

Ennegram. <http://www.9types.com/epd/1.php> - accessed 02 July 2013.

Epstein, Alex (Complications Ensur blog)
<http://complicationsensue.blogspot.co.uk/2006/11/polling-numbers.html> -
accessed 14 March 2014.

FarsiteForecast.com. <http://www.farsiteforecast.com/vanity-fair-polls-and-parties/> - accessed 18 March 2013.

Fischer, Gerhard. <http://l3d.cs.colorado.edu/~gerhard/papers/ccmcd2001.pdf> -
accessed 05 February 2014.

Fountain (John August). <http://www.fountain.io> - accessed 29 March, 2013.

Freeman, Eric T. <http://www.cs.yale.edu/homes/freeman/lifestreams.html> -
accessed 18 March 2014.

Giles and Anderson. <http://clearcongressproject.com/charviz/> - accessed 11 June 2013.

Gloor, Peter. <http://www.vanityfair.com/online/oscars/2013/02/mathematical-formula-oscar-winners-imdb> - accessed 08 March 2012.

Gold Light. <http://www.davekellyentertainment.com/marketsolution.html> - accessed 11 August 2007.

Google Ngram viewer. <http://books.google.com/ngrams/> - accessed 26 April 2013.

Grimes, Seth (2011). *12 Things the Semantic Web Should Know about Content Analytics*. <http://semanticnavigation.opentext.com/wp-content/uploads/2011/06/12-Things.pdf> - accessed 18 December 2013.

Hamlet social network:

<http://services.alphaworks.ibm.com/manyeyes/view/SJjqGFsOtha6ymEZWajPF2> - accessed 23 September 2008.

<http://www.jibble.org/shakespeare/images/hamlet.xml-00000384.png> - accessed 06 July 2008.

Hanser E., McKeivitt P., Lunney T., and Condell J. *Scenemaker: Intelligent Multimodal Visualisation of Natural Language Scripts*, in R. Setchi et al. (Eds.): KES 2010, Part IV, LNAI 6279, pp. 430–439, 2010.
<http://www2.denizyuret.com/bib/hanser/hanser2010scenemaker/hanser2010.pdf> - accessed 12 May 2014.

Gil S., Kuenzel L. and Suen C. *Extraction and Analysis of Character Interaction Networks From Plays and Movies*.

<http://www.stanford.edu/~cysuen/projects/GilKuenzelSuen-CharacterInteractionNetworks.pdf> - accessed 12 May 2014.

Henderson, Peter and De Silva, Nishadi (2006). *A Narrative Approach to Collaborative Writing: A Business Process Model* in *8th International Conference on Enterprise Information Systems (ICEIS), Paphos, Cyprus, 23 - 27 May 2006*.
http://eprints.soton.ac.uk/261806/1/Camera-ready_paper_499.pdf - accessed 12 May 2014.

Hollywood Reporter (Spec Scripts).

<http://fencingwiththefog.blogspot.ca/2006/01/screenplay-sales-statistics.html>

- accessed 18 March 2014.

Hornall Anderson touch walls.

<http://www.hornallanderson.com/casestudy/better-living-through-technology>

- accessed 02 April 2012.

Huntley, Chris. <http://dramatica.com/articles/how-and-why-dramatica-is-different-from-six-other-story-paradigms> - accessed 25 February 2013.

Jackson, Peter. <http://gointothestory.blcklst.com/2011/04/peter-jackson-on-screenwriting-process.html> - accessed 04 September 2012.

Jewell, Michael O., Lawrence, K. Faith, Tuffield, Mischa M., Prugel-Bennett, Adam, Millard, David E., Nixon, Mark S., schraefel, m.c. and Shadbolt, Nigel R. (2005) 'OntoMedia: An Ontology for the Representation of Heterogeneous Media' in *Multimedia Information Retrieval Workshop*, ACM SIGIR.

<http://image.ntua.gr/swamm2006/resources/paper12.pdf> - accessed 07 February 2007.

Jinhong, Shen, Seiya Miyazaki, Terumasa Aoki, and Hiroshi Yasuda. *Filmmaking Production System with Rule-Based Reasoning*. Dissertation. The University of Tokyo, 2003. http://sprg.massey.ac.nz/ivcnz/Proceedings/IVCNZ_66.pdf - accessed 05 February 2007.

Jinhong, Shen, Seiya Miyazaki, Terumasa Aoki, and Hiroshi Yasuda. *Filmmaking Production System with Rule-Based Reasoning*. Image and Vision Computing New Zealand, University of Tokyo.

http://sprg.massey.ac.nz/ivcnz/Proceedings/IVCNZ_66.pdf - accessed 07 February 2007.

Jurafsky, Shriberg, Fox and Curl. <http://acl.ldc.upenn.edu/W/W98/W98-0319.pdf> - accessed 29 March 2013.

Kao, Albert B. and Couzin, Iain D. (2014). *Decision accuracy in complex environments is often maximized by small group sizes*. Proceedings of the Royal Society B – accessed 23 April 2014.

<http://rspb.royalsocietypublishing.org/content/281/1784/20133305.abstract> - accessed 28 April 2014.

Kendall and Kendall. Slide 8-3 in <http://www.mgt.ncu.edu.tw/~ylchen/sasd-slide/chap08.ppt> - accessed 11 March 2014.

Kleinbard, Lily. *Adapting 'Life of Pi' for the Big Screen Took 170 Script Revisions*. <http://www.theatlantic.com/entertainment/archive/2012/11/adapting-life-of-pi-for-the-big-screen-took-170-script-revisions/265367/> - accessed 18 March 2014.

Korsakow system. <http://korsakow.org> - accessed 18 March 2014.

Maioriello, James.

<http://www.developer.com/design/article.php/1474561/What-Are-Design-Patterns-and-Do-I-Need-Them.htm> - accessed 03 January 2013.

Mairesse, F and Walker M. *PERSONAGE: Personality Generation for Dialog* in Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics, pages 496–503, Prague, Czech Republic, June 2007.

<http://acl.ldc.upenn.edu/P/P07/P07-1063.pdf> - accessed 12 May 2014.

McKee, Robert. <http://artscienceblog.blogspot.co.uk/2012/04/should-good-brand-stroy-chart-like-good.html> - accessed 11 June 2013.

McNiff, Jean. <http://www.jeanmcniff.com/ar-booklet.asp> - accessed 11 March 2014.

Media Technology (RCMT) and School of Creative Media, 2004.

<http://www.ieeesmc.org/announcements/Newsletter/Dec2004/SMCS-E-NewLetter-Liu2004.pdf> - accessed 05 February 2007.

MediaWiki Screenplay Extension <http://siege.org/projects/mwspe> - accessed 05 April 2006.

Montelbano Infographic. <http://cargocollective.com/federicafragapane/LA-LETTURA-CORRIERE-DELLA-SERA> - accessed 09 July 2013.

Moretti, Franco (2011). *Network Theory, Plot Analysis*. Literary Lab, Pamphlet 2, May 1, 2011. http://litlab.stanford.edu/?page_id=255 - accessed 31 December 2013.

Moubry, Sean. <http://screenplayXML.org> - accessed 10 June 2013.

Murtagh, Fionn, Ganz, Adam and McKie, Stewart (2008). 'The Structure of Narrative: the Case of Film Scripts' in *Pattern Recognition*, 42 (2), 302-312, 2009. <http://arxiv.org/abs/0805.3799> - accessed 24 May 2008.

Murtaugh, Michael. *The Automatist Storytelling System: Putting the Editor's Knowledge in Software*. Dissertation. MIT Media Lab., 1996. <http://xenia.media.mit.edu/~murtaugh/thesis/> - accessed 05 February 2007.

Myers, Scott. <http://www.gointothestory.com/2009/01/spec-script-sales-analysis-2008-big.html> - accessed 27 July 2009.

Narrativization - <http://www.narrativization.com> - accessed June 17, 2014.

Nature, vol. 453, p. 708, 4 June 2008. <http://www.nature.com/news/2008/080604/full/453708a.html> - accessed June 17, 2014.

Neumann, Petra, Stefan Schlechtweg, and Sheelagh Carpendale (2005). 'ArcTrees: Visualizing Relations in *Hierarchical Data*'. EUROGRAPHICS - IEEE VGTC Symposium on Visualization, 2005, IEEE. http://pages.cpsc.ucalgary.ca/~pneumann/wiki/papers/Neumann_2005_ATV.pdf - accessed 7 February 2007.

New York Times:

Candidates Debate.

(<http://www.nytimes.com/interactive/2007/12/15/us/politics/DEBATE.html> -
accessed on 06 July 2008.

Democratic Debate.

http://www.nytimes.com/interactive/2008/04/16/us/politics/20080416_DEBATE_GRAPHIC.html# - accessed 06 July 2008.

Storylines. <http://www.nytimes.com/2005/04/24/magazine/24TV.html> -
accessed 06 July 2008.

Ng, Kal, Marc Aurel Schnabel, and Thomas Kvan. *Architectural Animation becomes Alive*. Communicating Space(s) [24th eCAADe Conference Proceedings / ISBN 0-9541183-5-9] Volos (Greece) 6-9 September 2006, 598-603.

<http://wwwpeople.arch.usyd.edu.au/~marcaurel/publications/ecaadekal.pdf> -
accessed 07 February 2007.

Nike Touch Wall

<http://www.flickr.com/photos/davidjkelly/sets/72157624420150769/> -
accessed 02 April 2012.

Nitsche, Michael, and Maureen Thomas(2007). 'Stories in Space: the Concept of the Story Map' in *Visual Storytelling* Second International Conference, ICVS 2003, Toulouse, France, November 20-21, 2003, Proceedings Series: Lecture Notes in Computer Science, Vol. 2897 Balet, Olivier; Subsol, Gérard; Torguet, Patrice (Eds.)

http://www.lcc.gatech.edu/~nitsche/download/Nitsche_ICVS_03.pdf - accessed
07 February 2007.

The Numbers. <http://www.the-numbers.com> - accessed 27 July 2009.

The Object Management Group (OMG). <http://www.omg.org> - accessed 12
February 2014.

Optimizely <https://www.optimizely.com>, accessed July 09, 2014.

O'Reilly, Tim (2005). *What is Web 2.0: Design Patterns and Business Models for the Next Generation of Software*. O'Reilly Media.

<http://oreilly.com/web2/archive/what-is-web-20.html> - accessed 17 December 2013.

Paisa, Sabi. <http://scriptsuper.wordpress.com/2013/04/09/directors-homework/about> - accessed 17 June 2013.

Parham, Jason. <http://www.complex.com/tech/2013/06/amazon-storyteller-turns-scripts-into-storyboards> - accessed 09 July 2013.

Pate, John. *A LATEX2 ϵ class for Draft Screenplay*(2006).

<http://www.tug.org/texlive/Contents/live/texmf-dist/doc/latex/screenplay/screenplay.pdf> - accessed 23 January 2012.

Protovis. <http://mbostock.github.io/protovis/ex/> - accessed 11 June 2013.

Ramirez, Alejandro. *Towards a Cinematically Enhanced Narrative* in Proceedings of the 2005 ACM SIGCHI International Conference on Advances in Computer Entertainment Technology, 2005, ACM Press.

<http://delivery.acm.org/10.1145/1180000/1178550/p365-ramirez.pdf?key1=1178550&key2=7686480711&coll=&dl=acm&CFID=15151515&CFTOKEN=6184618> - accessed 07 February 2007.

Reconstruction Vol.8, No.3 (2008).

<http://reconstruction.eserver.org/Issues/083/mckie.shtml> - accessed June17, 2014.

Ruecker, Stan.

http://academia.edu/3155085/WATCH_MY_MOVES_FROM_DIGITAL_PLAYS_TO_THE_DIGITAL_PLAYBOOK - accessed 11 July 2013.

Schmidt, Benjamin (Argo).

<http://www.prochronism.com/2013/02/current-events-argo-anachronisms.html> - accessed 12 March 2013.

Screenwriting 2.0 in the Classroom? Teaching the Digital Screenplay.

<http://framescinemajournal.com/article/screenwriting-2-0-in-the-classroom/> - accessed 17 Dec. 2013.

ScriptReader. <http://scriptreaderonline.com> - accessed 03 September 2012.

Sentiment analysis:

Go, Bhayani and Huang.

<http://cs.stanford.edu/people/alecmgo/papers/TwitterDistantSupervision09.pdf> - accessed 11 June 2013.

Hennessey. J.W. (phpInsight class).

<https://github.com/JWHennessey/phpInsight> - accessed 18 June 2013.

Thet, Na and Khoo.

<http://jis.sagepub.com/content/36/6/823.full.pdf+html> - accessed 11 June 2013.

Yessenov and Misailovic.

<http://people.csail.mit.edu/kuat/courses/6.863/report.pdf> - accessed 11 June 2013

Scali, Gabriele and Graham Howard (2004). XML Coding of Dramatic Structure for Visualization. Dissertation. System Simulation Ltd., 2004.

http://www.spacespa.it/studi_ricerche/pdf/XML_Coding.pdf - accessed 01 February 2007.

Schilder <http://www.aclweb.org/anthology-new/W/W98/W98-0310.pdf> - accessed 29 March 2013.

Segel, Edward and Heer, Jeffrey. *Narrative Visualization: Telling Stories with Data*. <http://vis.stanford.edu/files/2010-Narrative-InfoVis.pdf> - accessed 18 March 2014.

Shakespeare:

Macbeth. <http://www.shakespeare-navigators.com/macbeth/Blood.html> - accessed 04 March 2013.

Open Source Shakespeare.org (<http://www.opensourceshakespeare.org/stats/> - accessed 04 March 2013.

DocuScope. <http://www.cmu.edu/hss/english/research/docuscope.html>) - accessed 04 March 2013.

Shakespeare's First Folio corpus <http://www.fastcompany.com/1800987/data-minings-thing-shakespeare-takes-center-stage-digital-age> - accessed 04 March 2013.

Shooting People (online community). <http://shootingpeople.org>.

SMAP (2007) <http://dl.acm.org/citation.cfm?id=1339060> - accessed June 17, 2014.

Soulier, Eddie and Caussanel (2002). *Narrative tools to improve Collaborative Sense-Making*. AAI Technical Report WS-02-09.

<http://www.aaai.org/Papers/Workshops/2002/WS-02-09/WS02-09-002.pdf> - accessed 7 February 2007.

Spitball. <http://spitball.com/2009/04/how-we-collaborate-on-a-screenplay> - accessed 28 August 2012.

Stapleton, Christopher, and Dana Mott. *Making Memories: Linear Versus Interactive Storytelling Collaboration in Mixed Reality Space*.

<http://www.rm.is.ritsumei.ac.jp/MR-PreVizProject/pdf/stapleton.pdf> - accessed

07 February 2007.

StoryVisualizer. <http://www.irit.fr/recherches/SAMOVA/ERCOLESSI/StoViz/#> - accessed January 15 2013.

Strapparava, Carlo and Mihalcea, Rada. *Learning to Identify Emotions in Text* in SAC'08 March 16-20, 2008, Fortaleza, Ceara, Brazil.

<http://web.eecs.umich.edu/~mihalcea/papers/strapparava.acm08.pdf> - accessed 12 May 2014.

Tauberer, Joshau (2006). *What is RDF?* XML.com

<http://www.xml.com/pub/a/2001/01/24/rdf.html> - accessed 10 November 2008.

TechAmerica Foundation. *Demystifying Big Data*.

<http://www.techamericafoundation.org/bigdata> - accessed 07 June 2013.

Tessman, Kent. <http://www.kenttessman.com/2012/02/open-screenplay-format> - accessed 28 August, 2012.

Velikovsky, J. T. *CREATIVE PRACTICE THEORY: The Model*,

<http://storyality.wordpress.com/creative-practice-theory/> - accessed 09 July 2013.

Walker M.A., Grace L.I., Sawyer J.E. *An Annotated Corpus of Film Dialog for Learning and Characterizing Character Style*. Proceedings of LREC 2012.

http://www.lrec-conf.org/proceedings/lrec2012/pdf/1114_Paper.pdf - accessed 12 May 2014.

Watching the Script. <http://digitalplaybook.humviz.org> - accessed 23 September 2008.

Watson, Mark. <http://www.markwatson.com/opensource/> - accessed 02 April 2012.

Wenger, Etienne. <http://wenger-trayner.com> - accessed 05 February 2014.

Writers Guild of America Schedule of Minimums (2004).

http://www.wga.org/subpage_writersresources.aspx?id=68 - accessed 22 December 2013.

Wordseye.

http://www.cs.columbia.edu/~coyne/images/wordseye_siggraph.pdf - accessed 07 February 2007.

XML formats - <http://screenplayxml.org/> <http://openscrfmt.sourceforge.net/> <http://freecode.com/projects/screenplay-xml> - accessed 23 January 2012.

XBRL example. <http://xbrl.squarespace.com/journal/2008/12/18/hello-world-xbrl-example.html> - accessed 6 June 2013.

Zhi-Qiang, Liu, and Leung Ka-Ming (2005). 'Script Visualization (ScriptViz): a Smart System That Makes Writing Fun in *Soft Computing* 10 (2005), 34-40. <http://www.springerlink.com/content/m1811w1261160627/fulltext.pdf> - accessed 07 February 2007.

Zhuang, Li, Feng Jing, and Xiao-Yan Zhu (2006). 'Movie Review Mining and Summarization' (2006) in CIKM '06 Proceedings of the 15th ACM international conference on Information and knowledge management, 43-50. http://www.csai.tsinghua.edu.cn/~zxy/paper/cikm249_zhuang.pdf - accessed 07 February 2007.

Appendix A: Hoffman's Hello World XBRL Example

This is a very simple example of using XBRL as provided by Charles Hoffman (aka the 'father' of XBRL) for learning purposes – see <http://xbrl.squarespace.com/journal/2008/12/18/hello-world-xbrl-example.html> - accessed 6 June 2013). Note that all data content is enclosed within <start> and </end > tags.HelloWorld Instance File

The initial lines of HelloWorld.xml provide links to essential reference files managed by the global XBRL authority at xbrl.org

```
<xbrl xmlns="http://www.xbrl.org/2003/instance"
xmlns:xbrli="http://www.xbrl.org/2003/instance"
xmlns:link="http://www.xbrl.org/2003/linkbase"
xmlns:xlink="http://www.w3.org/1999/xlink"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance"
xmlns:iso4217="http://www.xbrl.org/2003/iso4217"
```

The next lines refer to the online location of this specific instance file and the location of the XBRL schema this instance file references - HelloWorld.xsd. Typically, many instance files point to one schema file, which is why an XBRL schema can be considered 'mutually agreed'.

```
xmlns>HelloWorld="http://xbrl.squarespace.com/Hell
oWorld" xsi:schemaLocation=" ">
<link:schemaRef xlink:type="simple"
xlink:href="HelloWorld.xsd"/>
```

Then some contexts are defined for the data points listed in the file, which in this case include a reporting year, a reporting entity and a reporting period date:

```
<!-- Contexts -->
```

```

<context id="I-2007">
<entity>
<identifier
scheme="http://www.ExampleCompany.com">Example
Company</identifier>
<segment></segment>
</entity>
<period>
<instant>2007-12-31</instant>
</period>
</context>

```

In this instance, there are two contexts representing the reporting years 2006 and 2007. Then the reporting units are defined as monetary and denominated in US dollars:

```

<!-- Units -->
<unit id="U-Monetary">
<measure>iso4217:USD</measure>
</unit>

```

Now that the overall data context has been defined, the file lists a series of data points within that context and their respective attributes, known as 'facts', that comprise a title, a year, a decimals identifier and an amount (one fact each for reporting years 2006 and 2007):

```

<!-- Fact values -->
<HelloWorld:Land contextRef="I-2007" unitRef="U-
Monetary" decimals="INF">5347000</HelloWorld:Land>
<HelloWorld:Land contextRef="I-2006" unitRef="U-
Monetary" decimals="INF">1147000</HelloWorld:Land>

```

A number of facts are listed in the file, the above example simply reporting the data fact for 'Land' for each of the two context years: 2006, 2007.

The HelloWorld.xsd schema also references some key files like the instance file but mainly functions to define the behavior of the 'elements' that the 'fact' data is subject to, for example the element for the 'Land' fact:

```
<element  
name="Land"  
type="xbrli:monetaryItemType"  
substitutionGroup="xbrli:item"  
xbrli:periodType="instant"  
>
```

Appendix B: Selective Scenepad Feature List

Screenplay as Data

- All data and metadata stored in the MySQL database
- All views of script content 'assembled' by querying multiple, related tables
- Content views (PDFs) at script, scene, location (action) and role (dialog)
- Data granularity to snippet (i.e. dialog or action text) level
- Additional metadata for all entities (e.g. role gender, location/scene type, script genre etc.)

Screenplay Analytics

- Word frequency clouds derived from whole-script, dialog-only and action-only text
- Word frequency clouds based on parts of speech: Nouns and verbs
- Data and metadata-based charting at script, scene, location and role levels
- Timeline visualization of scene grouping at sequence and storyline levels
- 'Dial' visualization of script scenes by scene type metadata
- Character thrulines by character and role (e.g. protagonist/antagonist)
- Sentiment analysis of snippet content at script, action and dialog levels

Screenplay as Social Network

- Multi user application and ability to create script groups (i.e. communities of interest) to share scripts with

- Script sharing (e.g. private, group and public)
- Role access rights (i.e. read-only and read/write)
- Content hubs at script, scene, location and role levels for users to contribute images and links
- Scene commenting
- Access to a 'read-only' mobile site to review the script content either via a web link or a QR code
- link Scenepad entities (e.g. role or location) to images uploaded from a user's mobile device and stored on a third party website (e.g. Kee.ps.com or Photobucket.com)
- link Scenepad entities (e.g. script) to tweets posted to Twitter.com with the appropriate hashtag

Screenplay as Design

- Populate new blank script with sequence and role metadata based on Vogler's Hero's Journey
- Group scenes into structural devices such as sequences and storylines
- Attach images at script, scene, location and role levels to help to visualize the design of the screenplay

Business of Screenwriting

- Attach coverage reports to the script
- Post time and expense to a script development project
- Track script submissions to review sites and competitions
- Log ownership of script content at snippet level for authorship credits

Appendix C: Screenplay Analytics – A Use Case

Screenplay analytics of a datafied script enables the use of quantitative analysis, via machine-reading of the screenplay content, as a supplement and complement to the qualitative analysis that can be gained from human-reading of the script. Using my ScriptFAQ application (accessible at <http://www.screenplayanalytics/scriptfaq/login.php>) I have imported a Final Draft .fdx file in order to ‘datafy’ the screenplay content, by which I mean parsing the screenplay text in the file and ‘shredding’ it to store in a MySQL database.

Here I present a simple use case for how screenplay analysts/authors/readers can use screenplay analytics to help them understand the likelihood, not the certainty, that a script contains a strong protagonist by using basic screenplay analytics and by supplementing the screenplay data with additional metadata added by the reader/analyst. This analysis also shows how the addition of metadata (data about the screenplay data) by the analyst/author/reader can help to drive more analysis that may not be easy to get from the screenplay data alone.

I used ScriptFAQ to import and analyze a screenplay called *Reconciliation*. This is not a script that any reader will be familiar with as I wrote it and it is unpublished. But this unfamiliarity ensures that readers can follow the analysis below without any pre conceptions that may be present if a well-known movie script were used instead.

As soon as the script is imported we can go to the script ‘home’ page and see that some basic statistics about the screenplay content, namely:

- It has 71 Scenes involving 17 roles across 30 locations
- It has 1369 Snippets comprising 510 dialog and 342 action snippets

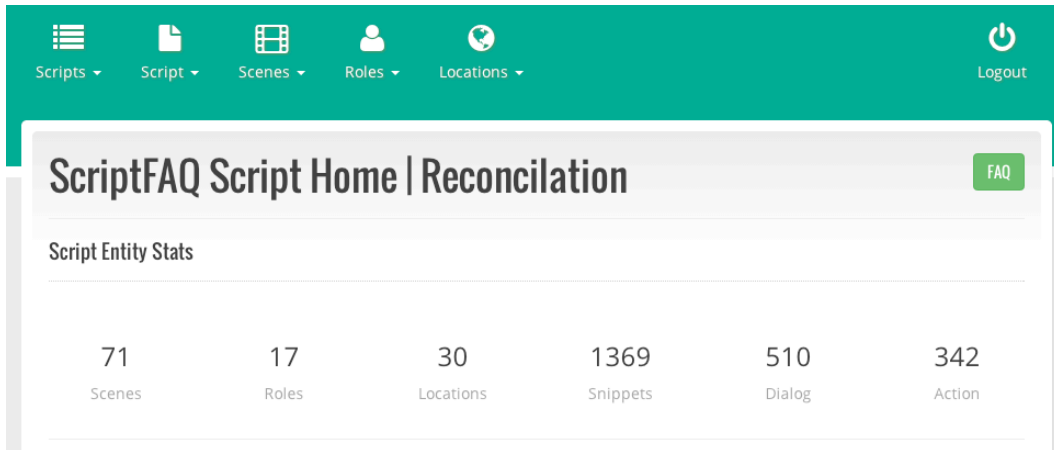


Figure C1 – Script Home Statistics

If we go to the role list we can see a list of roles identified from the script import. Notice some metadata – e.g. gender, type and logline – that would not have come from the script import but was added by the author later.

The "Script | Roles" page shows a list of 17 roles in Reconciliation. The table below displays the first 10 roles with their metadata:

#	Image	1 Cue	Gender	Type	Logline	Actions
1		AMERICAN	Male	Minor	US air force flyboy	
2		CLERK	Male	Minor		
3		DOREEN	Female	Antagonist	Land Girl - Sergeant's niece	
4		FRAU MANSTEIN	Female	Minor	Gerhard's wife in old age	
5		GERHARD	Male	Main	Downed German pilot	
6		GUARD	Male	Minor		
7		HAROLD	Male	Main	Local postman	
8		HERBERT	Male	Minor	Local retired diplomat	
9		HUSBAND	Male	Minor		
10		MAGGIE	Female	Protagonist	Managing family farm alone - husband is POW	

At the bottom of the table, there is a pagination control showing "10 25 50 100 All 1 2 Search", where "1" is highlighted in red.

Figure C2 – Script Roles

If we click the 'Home' icon next to the role called 'Maggie' we can already see some basic statistics about this role that may be interesting such as:

- The number of scenes and snippets the role is linked to
- The role rankings vs. other characters in the script

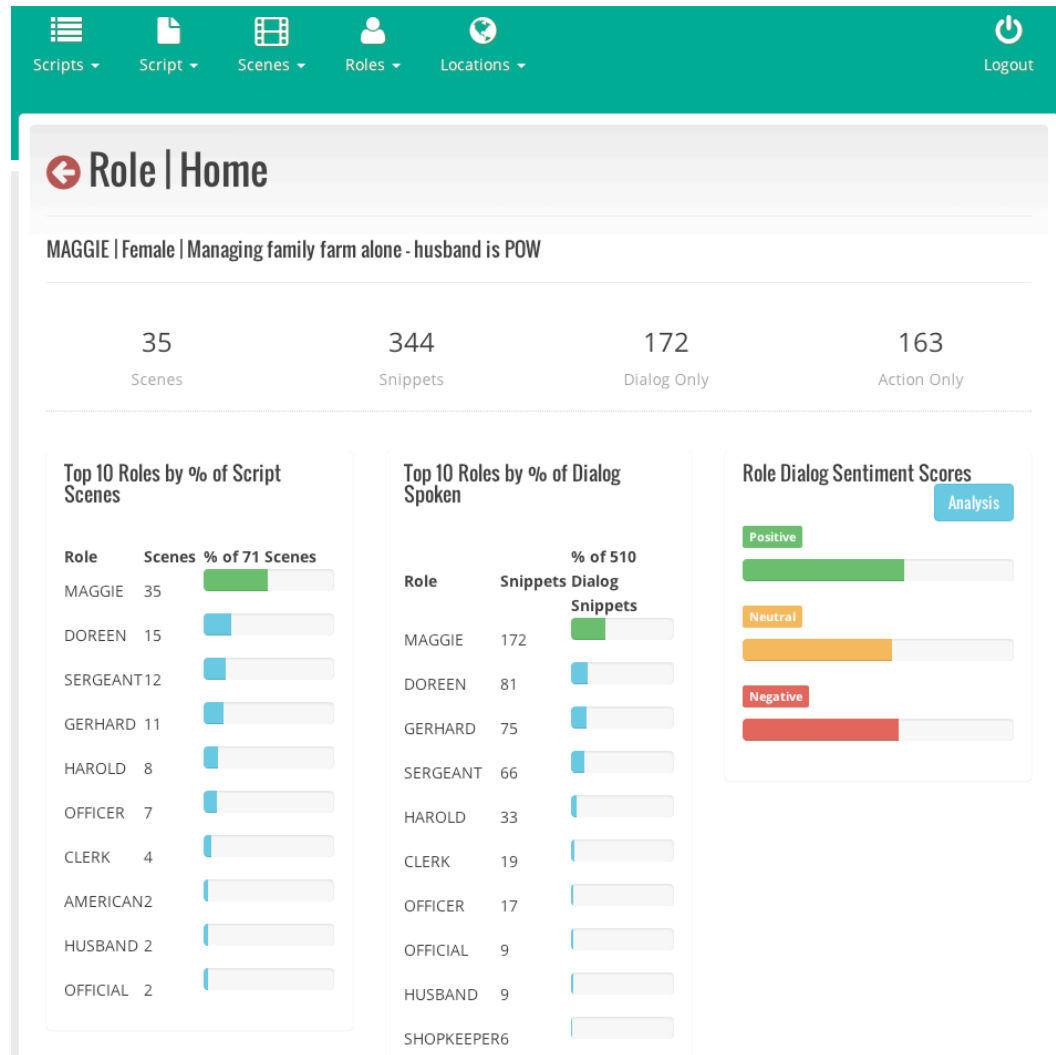


Figure C3 – Role Home Statistics

So of our 17 roles who looks most likely to be the protagonist? We go to the Role FAQs page to investigate.

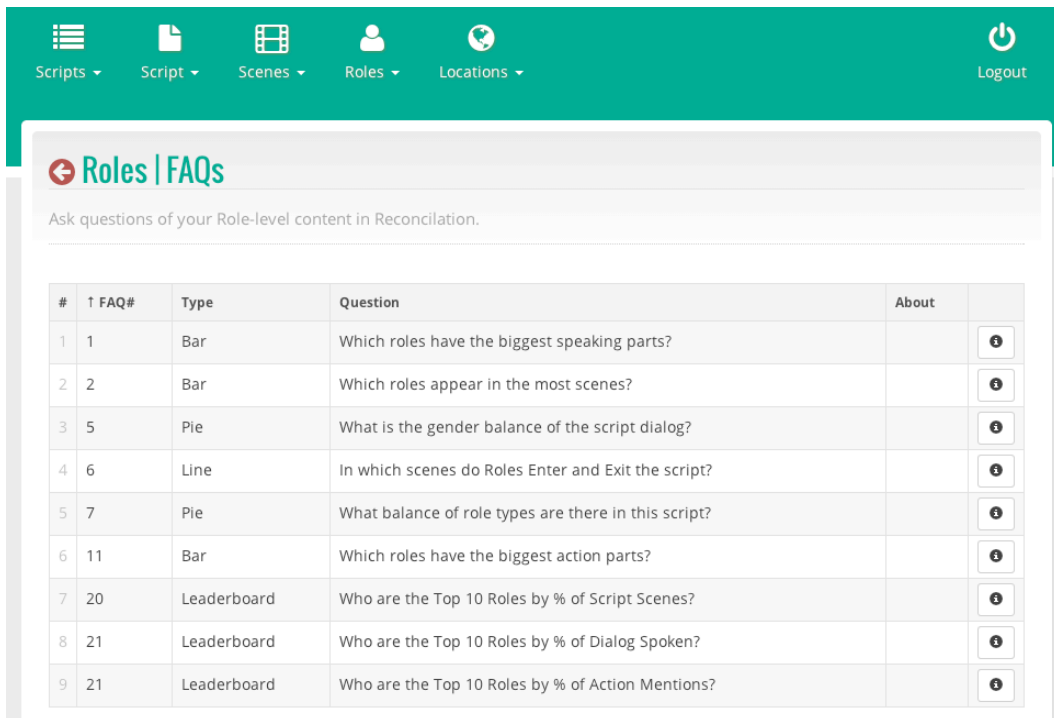


Figure C4 – Role FAQs

Here we see there are 9 Role FAQs to choose from – clicking the button on the right of the question displays an answer.

1. On-Screen Time

One indicator of a strong protagonist is likely to be a character that benefits from significant ‘on-screen’ time - in relation to other characters. On screen time is can be determined by at least three factors:

- The number of scenes the character appears in
- How much dialog they speak
- The ‘longevity’ of the character within the screenplay narrative

In the Role FAQ section, ScriptFAQ presents 3 simple charts to help understand on-screen time as shown in figures 5 and 6 below.

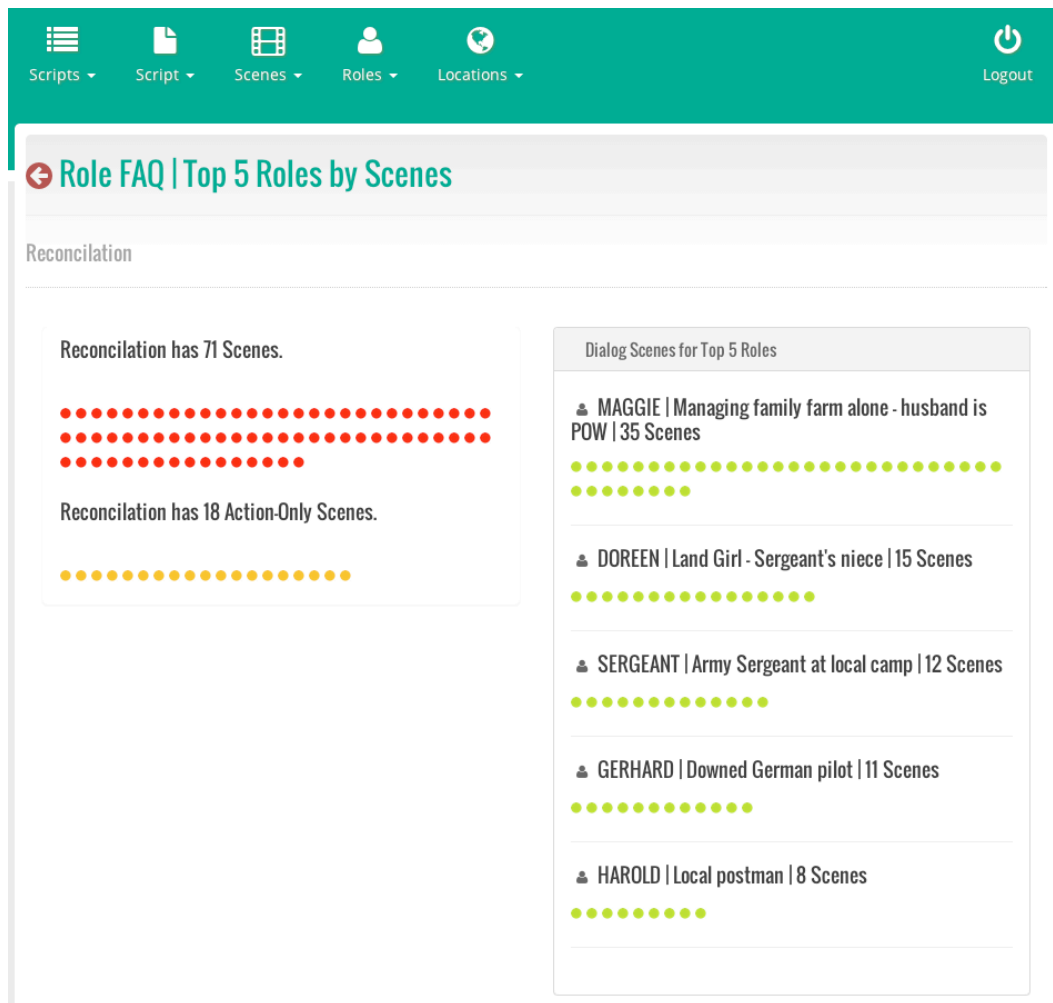


Figure C5 – Top 5 Roles by Scene (Note: each dot represents a unique scene)

In figure 5 we can see that the role 'MAGGIE' has a speaking part in a significant number of scenes (actually 35 out of 71). The next most significant character by this analysis is 'DOREEN' who only speaks in 15 scenes.

In figures 6 & 7 we can see that of the five major characters, Maggie enters the script first (in scene 4) and exits the script joint-last, in scene 69. So she enters the screenplay early and leaves late – both of which are likely to be good indicators of a protagonist role.

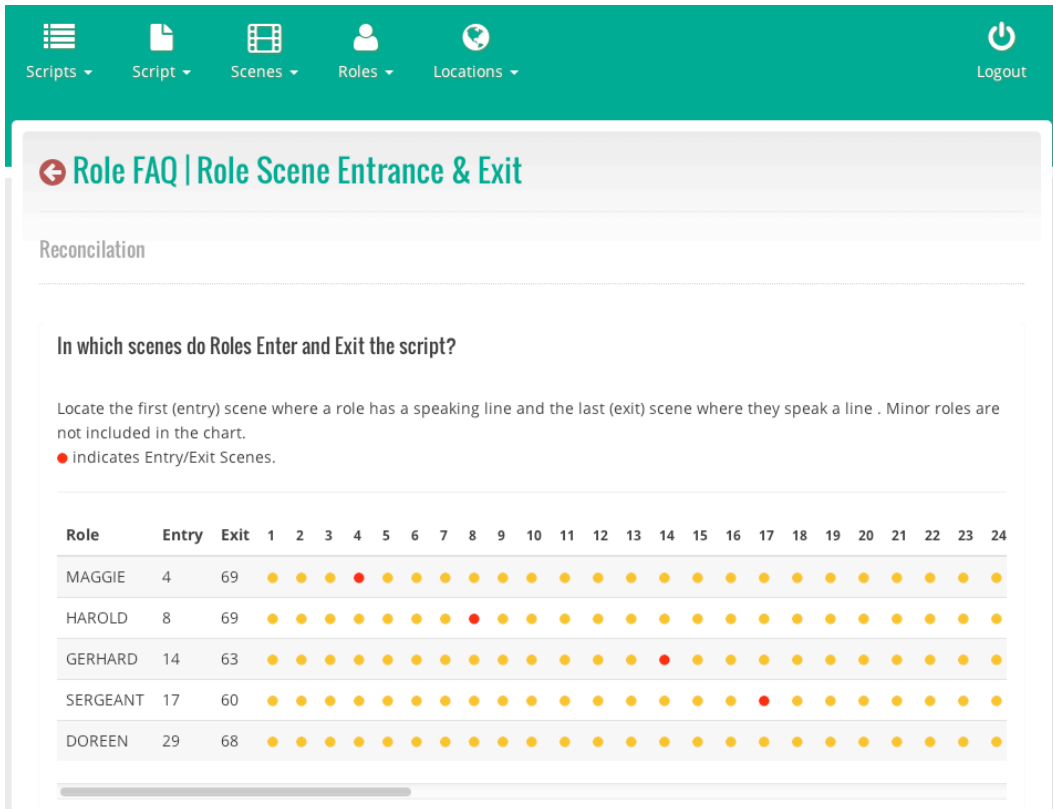


Figure C6 – Top 5 Roles by Entry Scene (Note: each dot represents a unique scene)

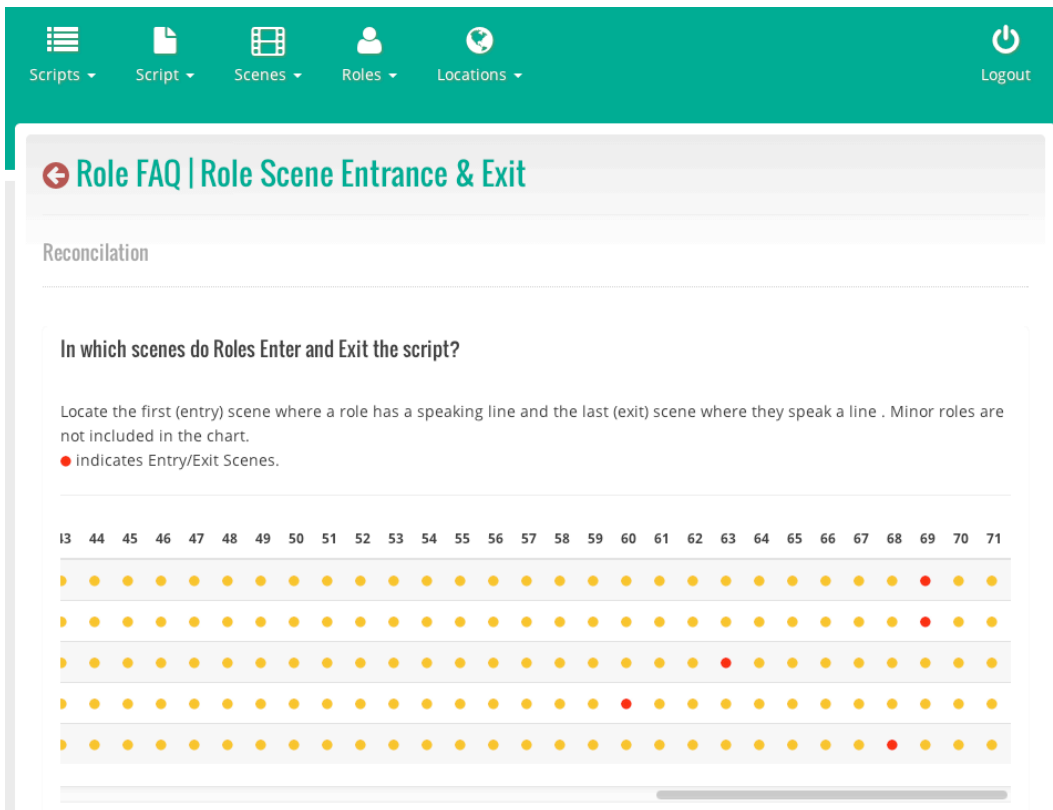


Figure C7 – Top 5 Roles by Exit Scene (Note: each dot represents a unique scene)

If you did not know that Maggie is the protagonist then you might already guess that she could be.

2. Dialog and Action

It's hard to imagine that a strong protagonist would not say and do a lot in a script. On screen time is unlikely to count for much if the character is mute and rooted to the spot as audiences are likely to become bored quite quickly with such a character. So in the Role FAQ section, as shown in figures 8 and 9 below, ScriptFAQ provides 2 more analyses that count how many dialog snippets a character speaks and an indication of how much action they participate in. In figure 8 we see that Maggie speaks 172 dialog snippets (of 510) and the next most vocal character appears to be Doreen (with 81 dialog snippets).

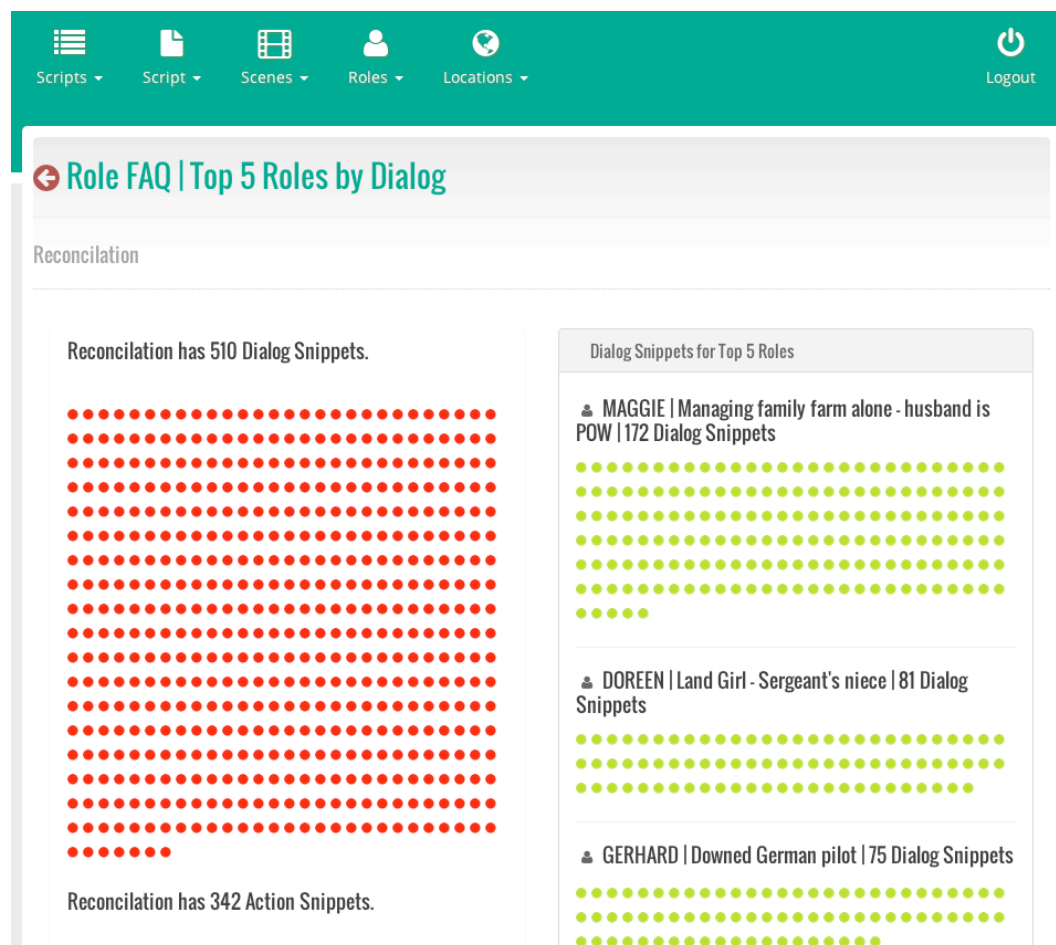


Figure C8 – Top 5 Roles by Dialog Snippets (Note: each dot represents a unique dialog snippet)

In figure 9 we see that Maggie’s role cue is mentioned in 163 action snippets and the next most ‘active’ character appears to be the Sergeant (mentioned in 47 action snippets). We know from figure 7 that the script as a whole contains 342 Action snippets, so Maggie is involved in almost half of them.

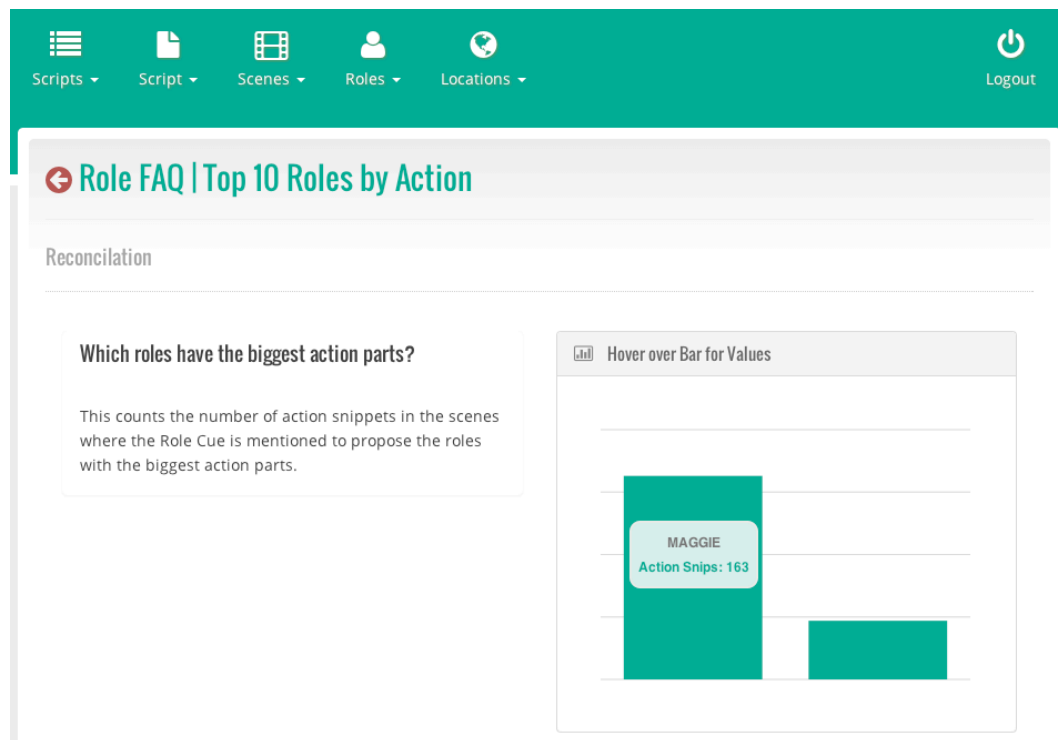


Figure C9 – Top 5 Role Cues mentioned in Action Snippets

Again, Maggie continues to exhibit ‘strong protagonist’ characteristics.

3. What The Protagonist Says and Does

Currently, ScriptFAQ does not provide any sophisticated textual analysis of the content of a character’s dialog and action snippets except for some basic part-of-speech analysis, of nouns and verbs in particular, that can provide some clues as to whether a character refers to potentially ‘interesting’ things or does/takes part in potentially ‘interesting’ action. In fact, figures 10 and 11 do not tell us that much but might if more sophisticated text analytics was available to apply to the script content.

← Role | Dialog

Top 20 Words/Nouns/Verbs in Reconciliation Dialog Snippets for Role MAGGIE

What are the Top Dialog Nouns and Verbs for this character?

List the most frequent nouns and verbs from the dialog snippets for the selected character.

Script Word Metadata

Settings for this script:

Top N Frequency = 20

Noun Hits >= 5

Verb Hits >= 3

Word Hits >= 3

Use default stoplist >= 1

Use custom stoplist >= 0

Custom stoplist words:

Only tested with English language scripts

Most Frequent Nouns/Verbs - By Dialog

Verb #	Noun #	Word #	#
leave 4	son 7	son 7	7
polish 3	harold 6	harold 6	6
run 3	home 6	home 6	6
	adlon 5	adlon 5	5
	camp 5	camp 5	5
	hotel 5	hotel 5	5
	man 5	killed 5	5
	time 5	man 5	5
	poland 4	time 5	5
	alright 3	german 4	4
	barn 3	leave 4	4
	doreen 3	poland 4	4
	hours 3	ready 4	4
	magazine 3	alright 3	3
	people 3	barn 3	3
	pilot 3	doreen 3	3
	room 3	hours 3	3
	uncle 3	lovely 3	3
		magazine 3	3
		people 3	3

Figure C10 - Most Frequent Nouns/Words in Dialog Snippets Spoken by Role

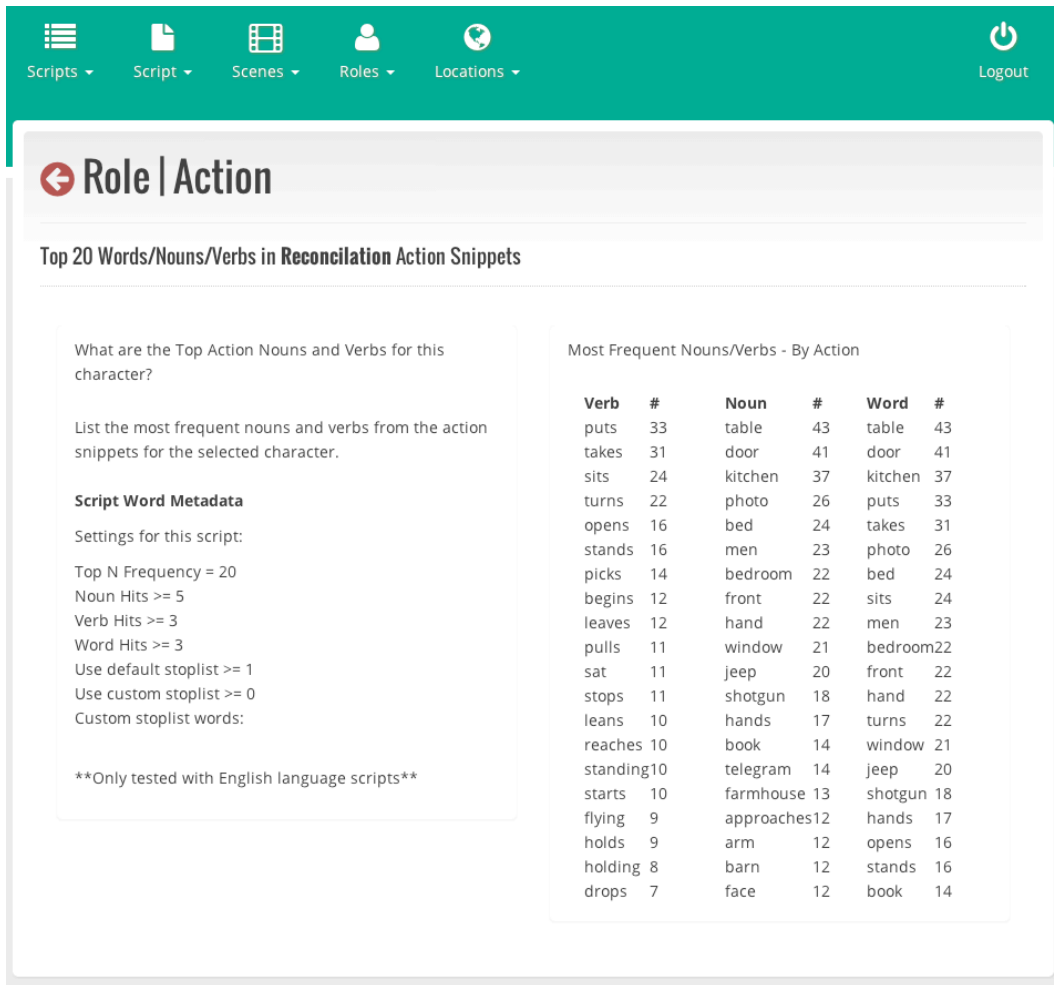


Figure C11 - Most Frequent Nouns/Words in Action Snippets of Role

4. Protagonist & Antagonist Role Metadata

Let's make the assumption that Maggie is the Protagonist. As the writer I know this already and so I would be expecting these kinds of results from my quantitative analysis of my script. If Maggie had not turned in these kinds of results I might have wondered why and whether in fact Maggie would be perceived by others (e.g. a reader/analyst or the potential audience of my movie) as the protagonist of this script.

So if I had not done so already, now would be the time to 'flag' Maggie as the protagonist by adding this metadata to her role record and while I'm at it I might add other metadata (e.g. her gender) as in figure 12 and probably do

the same job for the antagonist role.

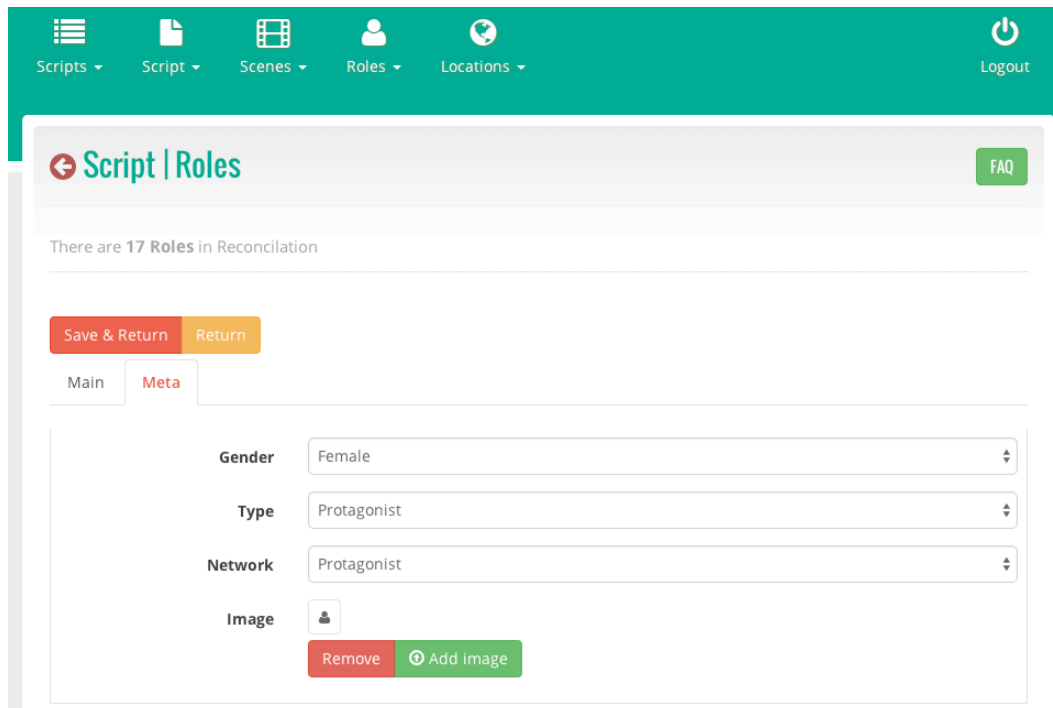


Figure C12 – Basic Role Metadata

Once you have identified your Protagonist and/or Antagonist in your script you can use another set of Key Role FAQs to analyze their relative dynamic.

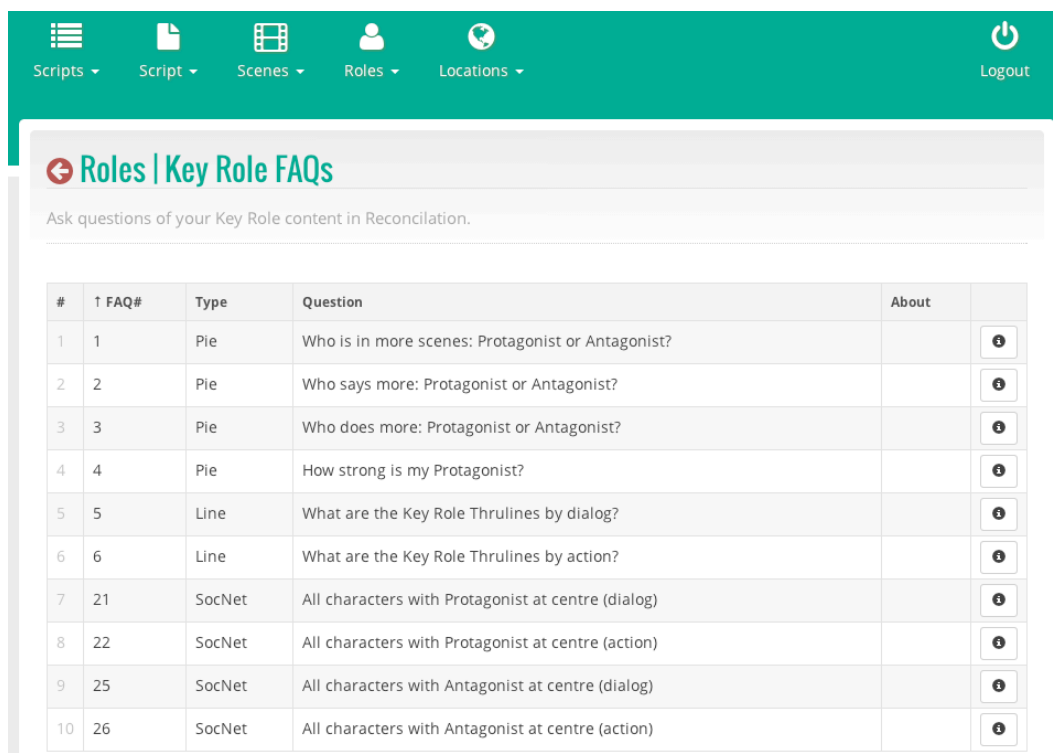


Figure C13 – Key Role FAQs

5. Protagonist Social Network

A strong protagonist is likely to have a rich set of social relationships in the script - this role in particular is unlikely to be 'isolated' from the rest of the cast. In other words the protagonist generally participates in an extensive social network. ScriptFAQ provides various analyses of role interactions as shown in figure 14 below. Here we see that Maggie (the 'central' role and identified by her 'protagonist' metadata) has dialog relationships with 10 other characters. Maggie is at the centre of a social network 'hub' where the thickness of each connecting 'spoke' line is indicative of the number of relationships she has with other characters in terms of the number of scenes in which both characters have a speaking role.

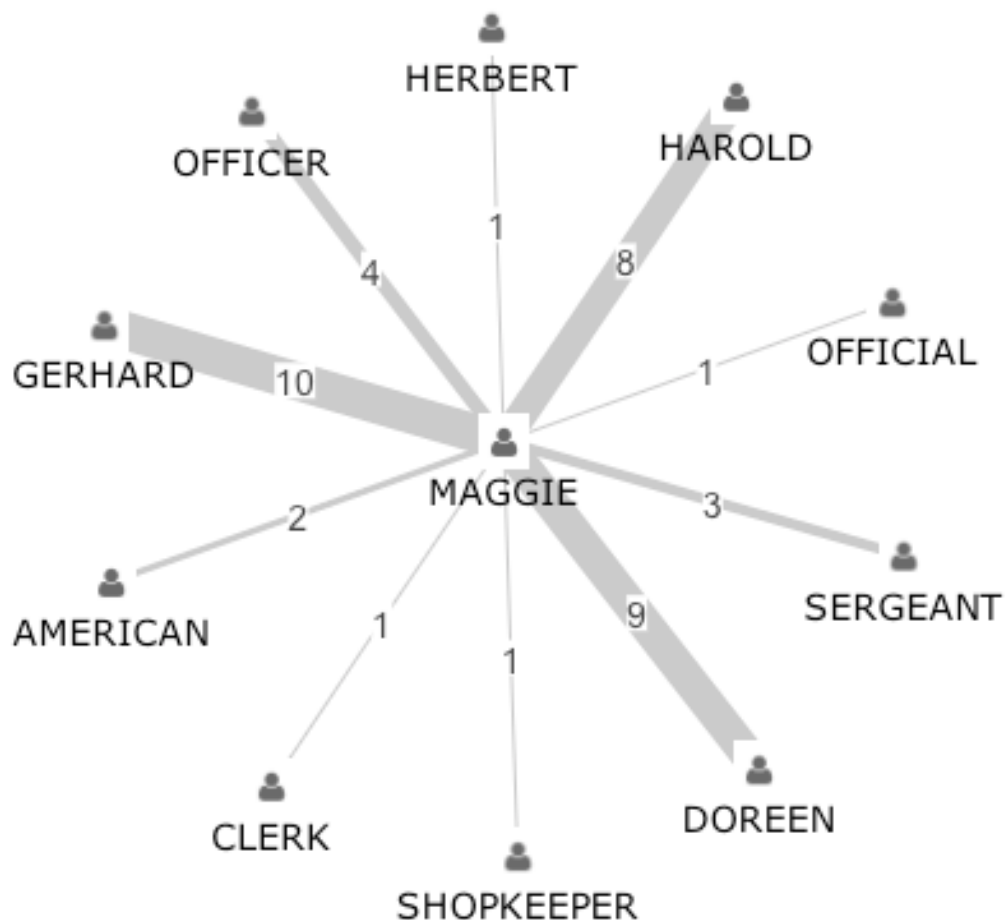


Figure C14 – Maggie as Social Network Hub

6. Protagonist Sentiment

ScriptFAQ does not use a particularly sophisticated sentiment analysis algorithm but it does provide some basic sentiment analysis to categorize role snippets as positive, neutral or negative. Conventionally, we might expect that our protagonist is generally more positive than negative and perhaps the reverse for our antagonist.

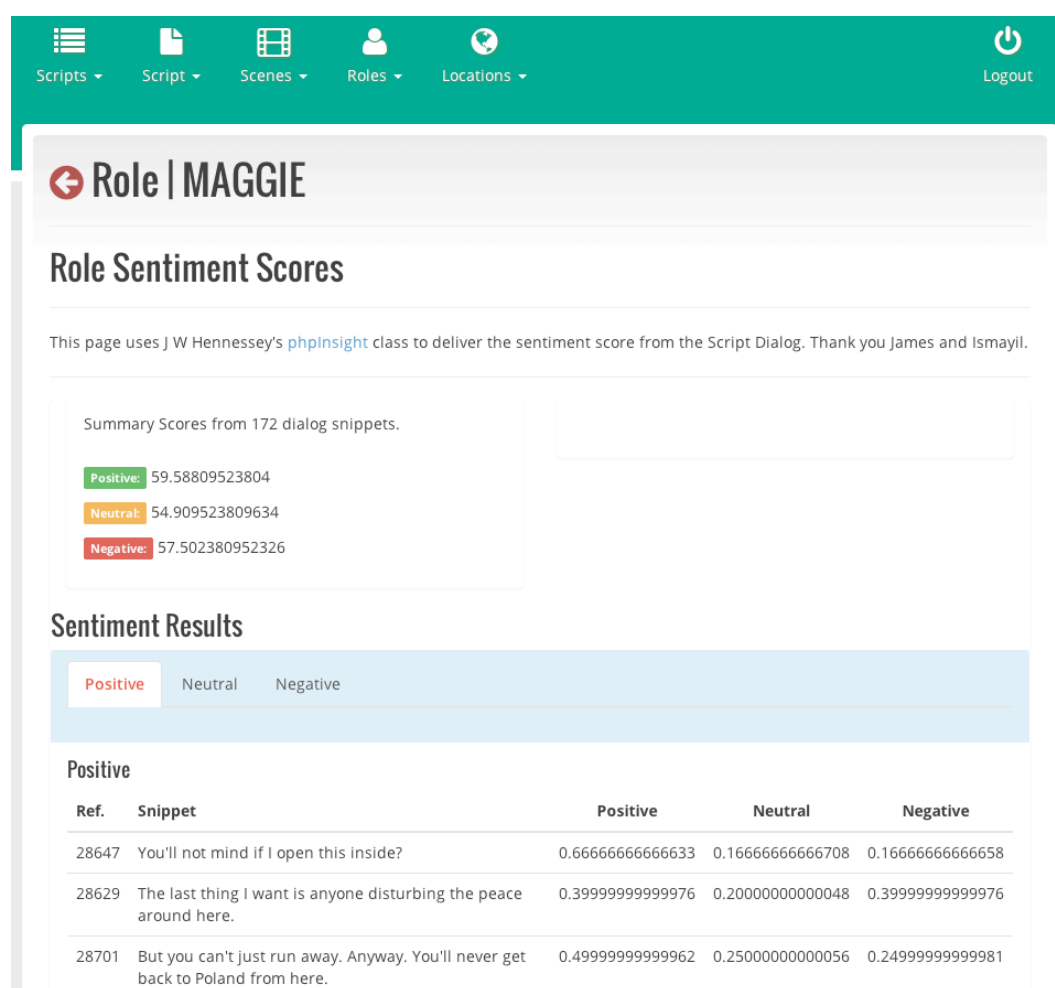


Figure C15 – Maggie's Sentiment Analysis

In figure 15 above we can see that Maggie's dialog is more positive than negative.

7. Protagonist vs. Antagonist

Some might argue that a Protagonist's strength can only be determined in relation to the script's Antagonist. This kind of analysis requires both the protagonist and antagonist character(s) to be identified by role metadata and that both can be mapped to a specific character. It may be that the protagonist and antagonist roles are in fact represented by more than one character. In this script, there is one character identified as the protagonist but two characters, loosely working in tandem, identified by metadata as the antagonist or antagonistic 'force'.

Figure 16 shows there is a roughly 70/30 'balance' in this script between scenes that the protagonist role(s) speaks in (35) and the antagonist role(s) speak in (15).

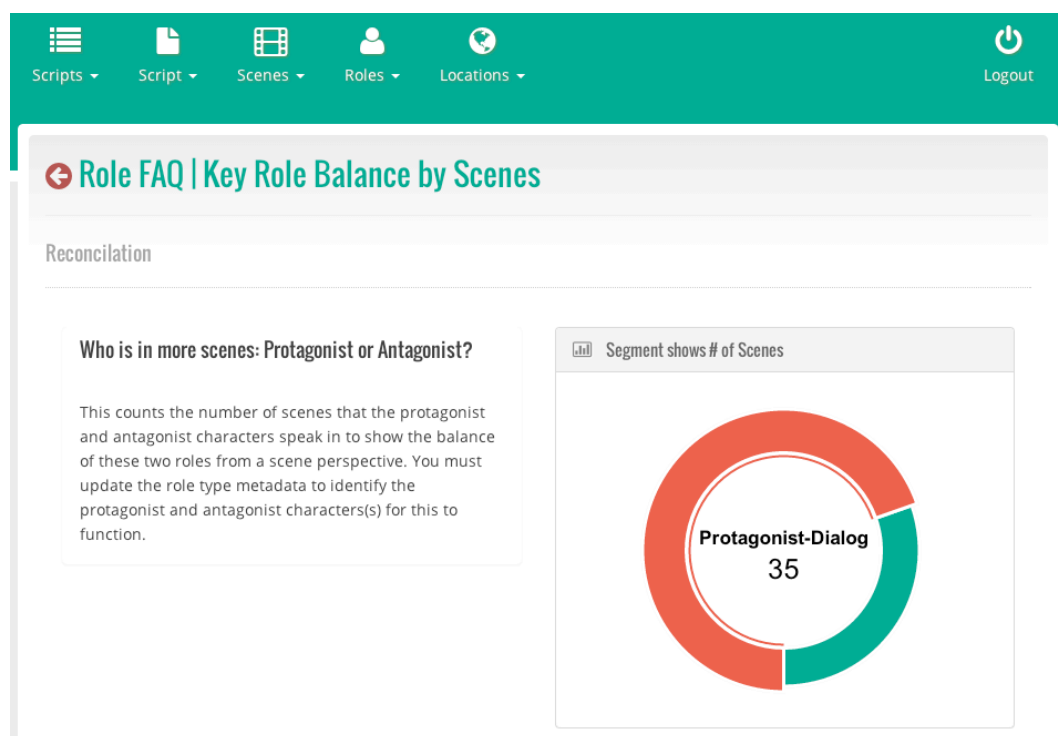


Figure C16 - Script Balance: Protagonist vs. Antagonist Dialog Scenes

Figure 17 shows who speaks more dialog snippets - protagonist or antagonist? In fact it's 172 vs. 81 again about a 70/30 balance.

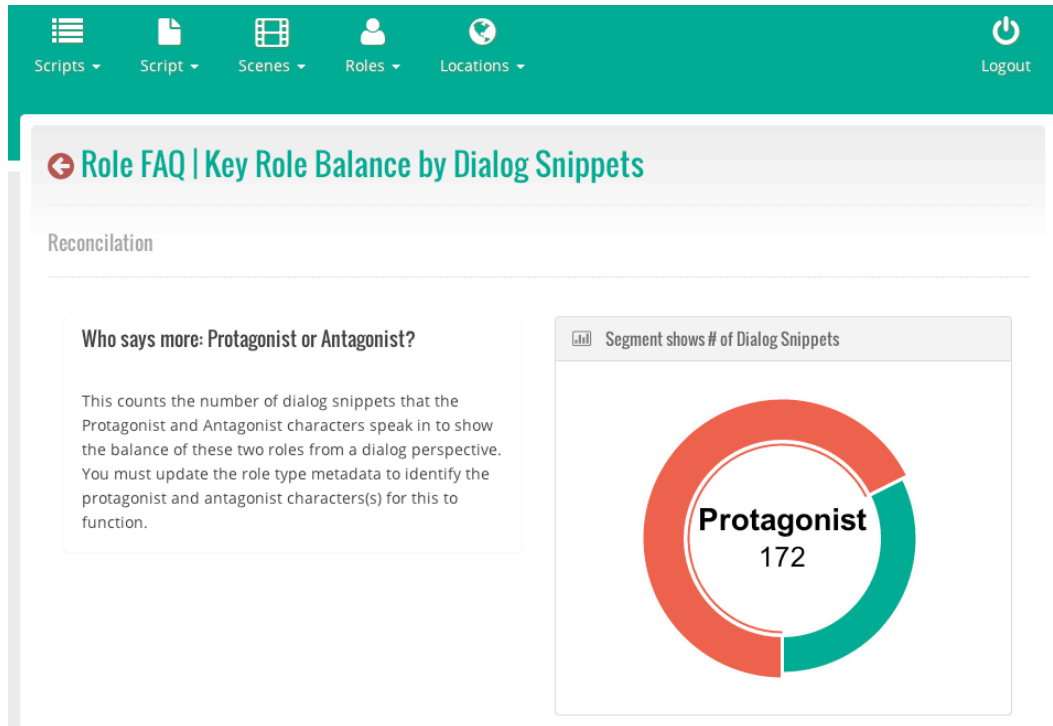


Figure C17 - Script Balance: Protagonist vs. Antagonist Dialog Snippets

Finally, figure 18 shows the balance of scenes that the protagonist speaks in in vs. scenes the rest of the cast speaks in but the protagonist does not (49/35).

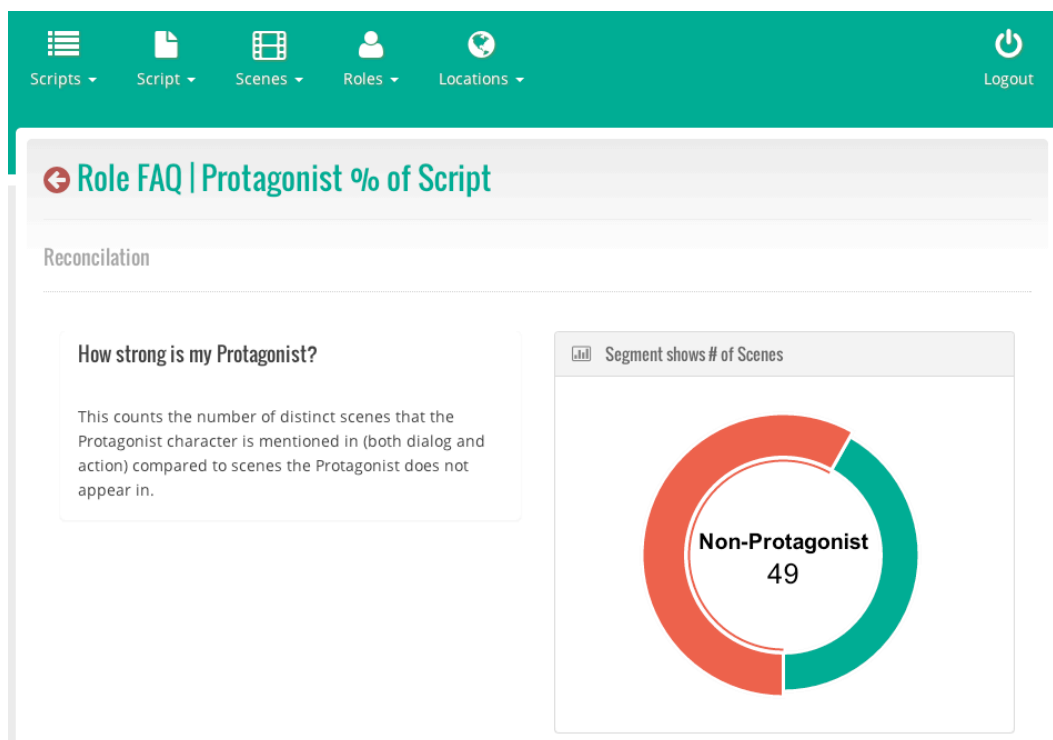


Figure C18 - Script Balance: Protagonist vs. Other Cast Scenes

We can also get some idea of the relative weight of the protagonist vs. the antagonist by comparing their social networks. Figure 19 shows the network for antagonist Doreen (vs. Maggie's network in figure 14 above).

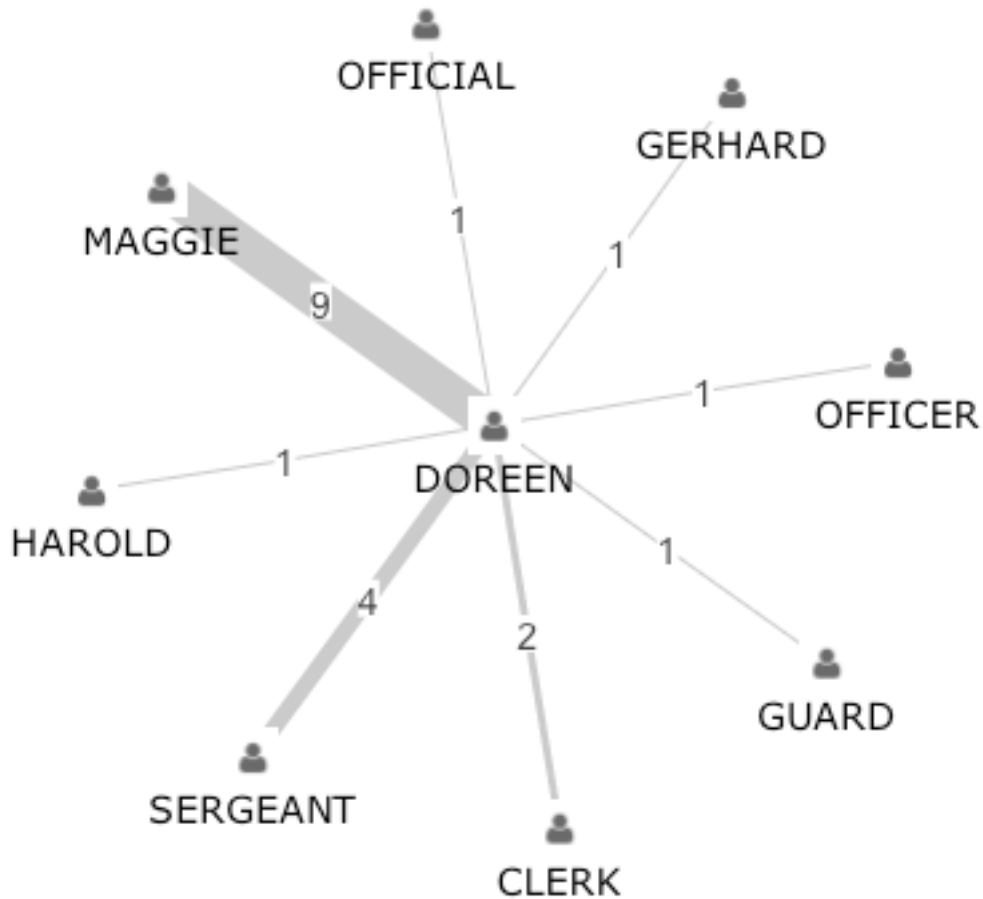


Figure C19 – Doreen (Antagonist) as Social Network Hub

Clearly, on the basis of these comparisons, Maggie remains a contender for classification as a strong protagonist.

Conclusion

Screenplay role analytics clearly has a long way to go, both in theory and in practice, but even these basic analyses show that you can learn something about the relative ‘strength’ of a protagonist even before you read the script. Assuming the screenplay is ‘datafied’, additional metadata must be added to the role record (in the database) to facilitate many of these analyses but this is a relatively trivial task for the script author to do. The figures above illustrate just some of the role-based analytics. ScriptFAQ also includes many more charts that analyze more about the script, its scenes and locations.

For example, *Reconciliation* looks like a good script for strong female roles since, as figure 20 shows, female roles speak slightly more dialog than male roles (264/246 snippets). In fact the script does have a very strong female protagonist so this is not surprising to me (the author).

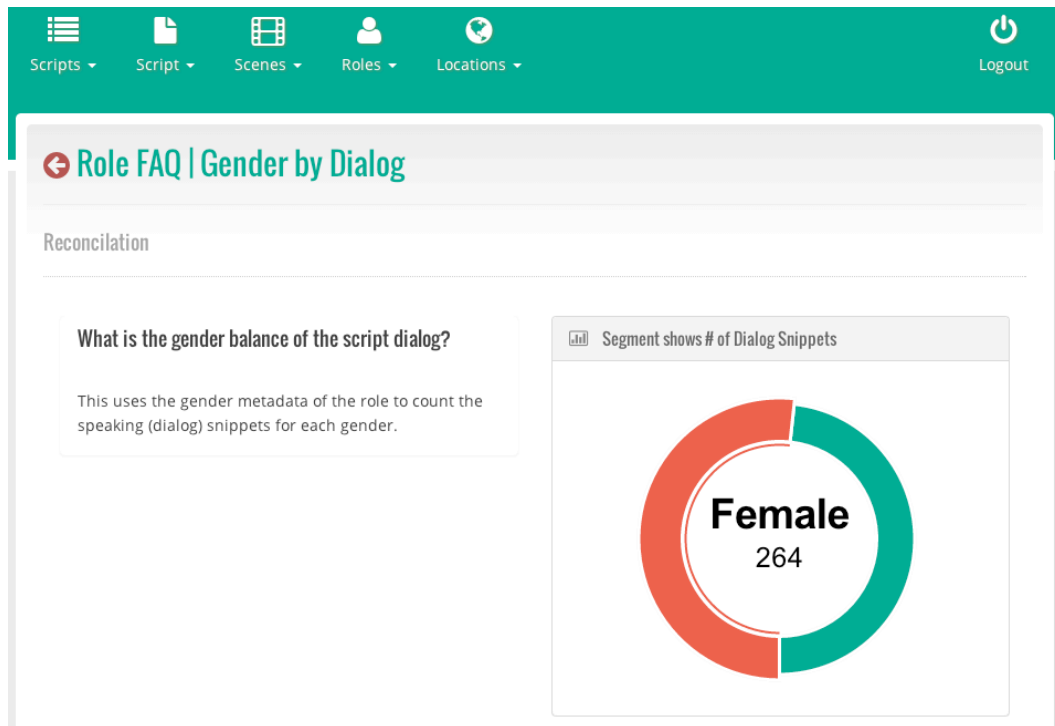


Figure C20 – Dialog Gender Balance

Of course it is your qualitative analysis from reading the script that will help to confirm or deny the validity of these quantitative analyses. This kind of

analytics correlates data; it does not indicate causality. A protagonist character who dominates the screen, relates to everyone and overpowers their antagonist with the weight of their dialog and action may still be flat and boring when it is clear, for example, exactly what they say and do and how they relate to others by reading the script. In fact their quantitative 'weight' may work against them when compared to your qualitative assessment.

I hope this simple analysis has shown that automated quantitative screenplay analytics, based on a datafied script, is a useful and interesting companion to traditional qualitative analysis via script reading.