

Behavioural Patterns for the Analysis of Creative Behaviour

Ph.D. Thesis

Keno Buss

This thesis is submitted in partial fulfilment of the
requirements for the degree of Doctor of Philosophy

Software Technology Research Laboratory (STRL)

De Montfort University

April 2011

Declaration

I declare that the work described in this thesis is original work undertaken by me between April 2007 and September 2010 for the degree of Doctor of Philosophy, at the Software Technology Research Laboratory (STRL), De Montfort University, United Kingdom. Apart from the stated degree, no other academic degree or award was applied for based on this work.

Publications

Keno Buss, Sascha Westendorf, Hussein Zedan, The Effect of Collaboration and Co-Creation on the Creative Processes, in Proceedings of The Second 2009 International Conference on Creativity and Innovations in Software Engineering (CISE'09: June 10-12, 2009, Ravda, Bulgaria)

Sascha Westendorf, Keno Buss, Hussein Zedan, Mining for Behavioural Knowledge and Information in the Creative Processes, in Proceedings of The Second 2009 International Conference on Creativity and Innovations in Software Engineering (CISE'09: June 10-12, 2009, Ravda, Bulgaria)

Hussein Zedan, Antonio Cau, Sascha Westendorf, Keno Buss, Andrew Hugill and Sue Thomas, Mapping Human Creativity, in Proceedings of the 12th Serbian Mathematical Congress, (28th August - 02th September, 2008, Novi Sad, Serbia)

Acknowledgements

I would like to express my gratitude to my supervisors Prof. Hussein Zedan, Prof. Sue Thomas and Prof. Stephen Brown for the guidance, support and advice during countless discussions. The completion of this thesis would not have been possible without their knowledge and wisdom.

I also would like to thank the members of the STRL and IOCT for their input and comments. Special thanks go to my colleague Sascha Westendorf, who has always been a great teammate and friend.

Furthermore, I would like to acknowledge the financial support from the De Montfort University and the IOCT. I appreciate the help of Prof. Andrew Hugill.

I'm especially thankful to my parents and family for their love, support and encouragement. They always believed in me and helped me to be where I am today. This thesis is dedicated to them.

Behavioural Patterns for the Analysis of Creative Behaviour

Abstract

Analysing human creativity has always been a difficult undertaking. The reason for this is the vagueness of the term creativity itself. Philosophers and Researchers have tried to unveil the secrets behind creativity since centuries. Besides some principles e.g. defined by Finke, Ward, and Smith [45], there is no holistic understanding about term *creativity* and there will always be discussions between researchers of different disciplines about how creativity is achieved.

Current approaches try to understand creativity through studying the creative artefact or the creative human himself. These approaches have not lead to any promising results. Instead of focussing on creativity itself, it is more promising to look at the creative process of creators. Through this, it is possible to analyse what happened during the creation of an artefact. The aim of the analysis is the identification of certain behaviours within the creative process, which lead to very creative results or in opposite hinder creativity. The gained knowledge is used for the creation of patterns, describing this specific behaviour. The analysis of the creative process is not only interesting for artists, but also for engineers, researchers, students and many others. Important target groups for this approach are schools and universities. Being able to early recognise problems in the learning curve of a student will enable a directed support, in order to improve the student's performance.

Until now, the main problem for analysing the creative process of a creator is the lack of information on how an artefact was created. Most creators tend not to make many notes during their creative phase, even if there are famous exemptions like Beethoven who left behind a huge amount of notices and marked changes in his work. Hence, the challenge is the development of a powerful framework, which can cope effectively with the recording and presentation of this creative process, in order to enable detailed analyses.

Modern computer technology fosters the recording of the creative process of a person. Much of today's creative work is achieved with computers and powerful software applications. Computer networks and the internet enable new ways of collaboration. The creation of a flexible, collaborative tool-set is the ideal approach for the mapping and analysis of the creative process.

A novel approach, based on these ideas, was developed by the members of the Creative Technologies Research Programme at the Software Technology Research Laboratory (STRL). The central aim of the approach is the mapping of the human creativity with so-called creativity maps. A creativity map is basically transition system, which allows to store and represent the creative process as well as to hold each version of the artefact. The developed De Montfort Creativity Assistant (DMCA), is a collaborative, web-enabled state-of-the-art software framework, realising the theoretical concepts.

This thesis represents a substantial contribution to the research project. The focus of the presented work lies on the analysis and the support of the creative process. Several novel techniques for the analysis of massive data sets have been developed. The presented techniques enable a domain-independent analysis and support of the creative process. This is huge improvement over previous concepts, which are highly specialised and focus on the creative artefact itself. The presented approach required the development of several novel techniques.

The major contributions of the presented research are:

- Behavioural Patterns
Enable the description of constructs inside the creativity map. These constructs represent the creative behaviour of the creator.
- Techniques for a Computer-supported Information Extraction
A pattern description language enables the computer-supported information extraction from creativity maps.
- Knowledge Repository
Knowledge, gained through the analysis processes, is stored centrally and shared with other creators in order to stimulate further growing of knowledge.
- Concepts for the Analysis and Support of the Creative Process
Novel concepts allow the structured analysis and support of the creative process, regardless of the creator's domain.

All developed techniques have been implemented in several tools, which aim to enrich the DMCA with the ability of a computer-supported analysis and support of the creative process.

Table of Contents

Acronyms	xv
1 Introduction	1
1.1 Motivation	2
1.2 Research Questions	3
1.3 Original Contributions	4
1.4 Scope of Thesis	5
1.5 Research Methodology	6
1.5.1 Selection of Background Material	6
1.5.2 Research Method	6
1.5.3 Data Collection	10
1.5.4 Analysis Technique	11
1.6 Thesis Structure	11
2 Background and Related Work	14
2.1 Introduction	15
2.2 Creativity and the Creative Process	15
2.2.1 Creative Process Models	18
2.2.2 Creativity Models	20
2.2.3 Creative Thinking Tests	22
2.3 Collaborative Creativity	24
2.3.1 Avoidance of Barriers in Creative Collaboration	24
2.3.2 Multimedia Literacy and Transliteracy	26

2.3.3	Computer Supported Cooperative Work and Groupware	27
2.3.4	Computer Supported Collaborative Learning	28
2.4	Creativity Support Tools	29
2.4.1	Impact of New Technologies on Creativity	31
2.4.2	Human-Computer Interaction	32
2.4.3	Examples of Creativity Support Tools	33
2.5	Limitations of Current Creativity Support Tools	34
2.6	Knowledge Management	36
2.6.1	Data Preparation and Retrieval	37
2.6.2	Conclusion	40
2.7	Summary	41
3	Preliminaries	43
3.1	Introduction	44
3.2	Mapping Human Creativity	44
3.2.1	Axioms of Creativity	45
3.3	Generalised Transition System	46
3.4	States	47
3.5	Transitions	48
3.5.1	Activity Types	49
3.5.2	Start and End of a Transition	50
3.5.3	Scope of Influence	51
3.5.4	Special Transitions	52
3.6	Operations	54
3.6.1	Choices	54
3.7	Sequences	56
3.7.1	Sequential Composition	56
3.7.2	Creative Path	57
3.8	Laws of Creative Processes	58

3.8.1	Identity and Units	58
3.8.2	Zeros	58
3.8.3	Commutativity	58
3.8.4	Associativity	58
3.8.5	Distributivity	59
3.8.6	Star and General Rules	59
3.9	Viewpoints	59
3.10	Creativity Maps	61
3.10.1	Semantics of the Creativity Map	62
3.10.2	Partial Creativity Maps	65
3.10.3	Uninfluenceable Activities	71
3.11	Summary	72
4	Pattern-Based Information Extraction	74
4.1	Introduction	75
4.2	Data, Information and Knowledge	75
4.3	Scope of the Research	76
4.4	Creative Behaviour	77
4.5	Behavioural Patterns	78
4.5.1	Length of Sequences	79
4.5.2	Length of Results	79
4.5.3	Complexity of Patterns	80
4.6	Behavioural Pattern Description Language	81
4.6.1	Operators and Formal Definition	82
4.6.2	The REVERT() Operator	90
4.6.3	Parsing Limitations	92
4.7	Application of Patterns	92
4.7.1	Sequence Patterns	93
4.7.2	Branching Patterns	95

4.7.3	Merging Patterns	97
4.7.4	Revert Patterns	100
4.8	Refinement Through Partial Creativity Maps	102
4.9	Summary	105
5	Knowledge Management and Store	107
5.1	Introduction	108
5.2	Knowledge Repository	108
5.2.1	Knowledge Categories for Behavioural Patterns	110
5.2.2	Metainformation	112
5.2.3	Structure of the Knowledge Repository	113
5.2.4	Knowledge Database	115
5.2.5	Knowledge Management Facility	116
5.2.6	Access Management	119
5.3	Knowledge Exchange	120
5.3.1	Search and Comparison	122
5.4	Summary	124
6	Analysis and Support of the Creative Process	125
6.1	Introduction	126
6.2	Analysis Process	126
6.2.1	Detailed Analysis	128
6.2.2	Monitoring of the Creative Process	130
6.2.3	Analysis of Frequent Behaviours	134
6.3	Hopping	136
6.4	Comparing Creativity Maps	137
6.4.1	Automated Comparisons of Creative Behaviour	139
6.5	Analysis of Collaboration	139
6.6	Creativity in the Digital and Non-Digital Age	140

6.7	Evaluation of Creative Behaviours	142
6.7.1	Statistics and Probabilities	142
6.8	Implications of the Analysis and Evaluation	144
6.9	Creativity Support	145
6.10	Summary	146
7	Tool Support	148
7.1	Introduction	149
7.2	De Montfort Creativity Assistant	150
7.2.1	Layers of the DMCA	152
7.3	Revert Tracking	155
7.3.1	Identification Technique	156
7.4	BPDL Parser	159
7.4.1	Structures and Preparation	160
7.4.2	Logical Representation of BPDL Expressions	163
7.4.3	Thompson's Construction	165
7.4.4	Subset Construction	169
7.4.5	Usage of the Automaton	173
7.4.6	Advanced Operations	176
7.5	Monitoring Process	180
7.6	De Montfort Creative Environment	182
7.6.1	Project Management Facility	183
7.6.2	Collaborative Editor	184
7.6.3	Text Analysis Facilities	186
7.6.4	Supporting Tools	187
7.7	De Montfort Creativity Mapper	190
7.7.1	Creativity Map Viewer	191
7.8	Knowledge Management and Analysis Tool	193
7.9	Summary	196

8	Case Studies	198
8.1	Introduction	199
8.2	Case Study 1 - Participants of a Writing Workshop	199
8.2.1	Comparison of Creative Behaviour	200
8.2.2	Conclusion	205
8.3	Case Study 2 - Self-Observation	206
8.3.1	Analysis of Changes in Creative Behaviour	206
8.3.2	Conclusion	210
8.4	Case Study 3 - The Musician	210
8.4.1	Background	211
8.4.2	Analysis of the Creativity Map	212
8.4.3	Analysis of the Partial Creativity Map	214
8.4.4	Conclusion	216
8.5	Summary	217
9	Conclusion and Future Work	219
9.1	Introduction	220
9.2	Conclusion	220
9.2.1	Review of the Research Questions	221
9.3	Review of the Research Methodology	223
9.3.1	Adequacy of the Research Methodology	223
9.3.2	Steps in Constructive Research	224
9.3.3	Choice of Case Studies	225
9.4	Critical Remarks	225
9.4.1	Data Collection Method	226
9.4.2	Validity of the Analyses	226
9.5	Future Work	227
9.5.1	Further Data Collection	227
9.5.2	Additional Capturing Techniques	227

9.5.3	Support of Novel Platforms	228
References		229
A Implementation		240
A.1	Knowledge Repository	240
A.1.1	Knowledge Database	241
A.1.2	Knowledge Management Facility	243
A.1.3	Revert Storing and Retrieving	245
A.2	BPDL Parser and Monitoring Process	248
A.3	Networking of the De Montfort Creativity Assistant	250
A.3.1	Data Transmission over Networks	250
A.3.2	Java Web Start	251
B Creativity Maps of Creative Writers		254
C Creativity Maps for Parts of the Thesis		264
D Creativity Map and Partial Creativity Map of the Musician		267

Acronyms

AI artificial intelligence

BPDL Behavioural Pattern Description Language

CSCL Computer Supported Collaborative Learning

CSCW Computer Supported Cooperative Work

CAT Consensual Assessment Technique

CST Creativity Support Tools

DBMS database management system

DFS Depth-first search

DMCA De Montfort Creativity Assistant

DMCE De Montfort Creative Environment

DMCM De Montfort Creativity Mapper

DFA deterministic finite automaton

EBNF Extended Backus-Naur Form

ERM entity-relationship model

GTS generalised transition system

GUI Graphical User Interface

HCI human-computer interaction

HTML Hypertext Markup Language

HTTP Hypertext Transfer Protocol

IP Internet Protocol

JAR Java Archive

JNLP Java Network Launching Protocol

JVM Java Virtual Machine

KMF knowledge management facility

NAT network address translation

NFA nondeterministic finite automaton

PCM partial creativity map

RDBMS relational database management system

RMI Remote Method Invocation

SOM Self-Organizing Map

SQL Structured Query Language

STUN Session Traversal Utilities for NAT

STRL Software Technology Research Laboratory

TCP Transmission Control Protocol

TTCT Torrance Tests of Creative Thinking

UML Unified Modelling Language

WSD word sense disambiguation

XML Extensible Markup Language

List of Figures

2.1	The Directed Creativity Cycle	19
2.2	The Systems View of Creativity	22
3.1	A Transition System	46
3.2	A Generalised Transition System	47
3.3	State	47
3.4	States of different Projects	48
3.5	Transitions between States	48
3.6	Transitions and States	51
3.7	Scope of influence	52
3.8	ϵ -transition	53
3.9	Branching	55
3.10	Merging	56
3.11	A Creative Path	57
3.12	Colour-coded Transitions	60
3.13	A Revert Action	63
3.14	A Creativity Map	64
3.15	Partial Creativity Map based on one Viewpoint	67
3.16	Partial Creativity Map based on two Viewpoints	68
3.17	Alternating Transitions	69
3.18	Creativity Map Example	69
3.19	Partial Creativity Map	70
3.20	Part of a Creativity Map with Parallel Transitions	71

4.1	Knowledge Extraction	75
4.2	Creativity of Leonardo da Vinci (1452-1519)	77
4.3	Graphical Representation of a Behavioural Pattern	78
4.4	Length of the Pattern	80
4.5	A long Sequence	80
4.6	Reverting to an old State	90
4.7	Revert Pattern	91
4.8	A Creativity Map Example	93
4.9	The Result of the Creation Pattern	94
4.10	Example for the Application of the Branching Pattern	95
4.11	Results of the Branching Pattern	96
4.12	Choice Pattern I	97
4.13	Results of the Merging Pattern	98
4.14	Choice Pattern II	99
4.15	Revert Example	100
4.16	Results of the Revert Pattern	101
4.17	Results of the second Revert Pattern	102
4.18	A Creativity Map	103
4.19	A Partial Creativity Map	104
4.20	Results of the Extraction Process	104
5.1	Exchange of Knowledge	109
5.2	Hierarchy of Patterns	111
5.3	The Knowledge Repository	114
5.4	Elements of a Knowledge Entry	115
5.5	Cycle of Knowledge Management	117
5.6	Personal Knowledge Access	119
5.7	Personal Knowledge List	121
5.8	Synonymy of Activities	123

6.1	Analysis Process	127
6.2	Monitoring of Current Behaviour	133
6.3	Hopping between Creativity Maps	136
6.4	Cycle of Analysis and Perception	142
6.5	Result of the Extraction Process	143
6.6	Probability of choosing a Transition	144
7.1	Layers of the DMCA	152
7.2	Example of Revert Operations	155
7.3	Creativity Map Example	161
7.4	Parse Tree	163
7.5	Basic Constructs for the NFA	165
7.6	Basic Constructs of the NFA Example	167
7.7	Sequence Construct of the NFA Example	167
7.8	Repetition Construct of the NFA Example	168
7.9	Choice Construct of the NFA Example	168
7.10	Input for the Subset Construction	171
7.11	Result of the Subset Construction	172
7.12	Creativity Map Example	173
7.13	Algorithm for the Traversing of the Creativity Map	174
7.14	Sub-routine for the Identification of Behavioural Patterns	175
7.15	Two Entities with Branches	178
7.16	Monitoring of Current Behaviour	181
7.17	Project Management Facility	183
7.18	Collaborative Editor	184
7.19	Collaborative Editor - Revision History	186
7.20	Access Control Management Facility	187
7.21	Instant Messaging Facility	188
7.22	Pend-it Notes	189

7.23 Creativity Mapper	190
7.24 Creativity Mapper Configuration Dialogue	191
7.25 Creativity Map Viewer	192
7.26 Knowledge Management	194
7.27 Analysis Facility	195
8.1 Results for the Creation Pattern	201
8.2 Description of the Labels	202
8.3 Results for the Revert Pattern - Part 1	203
8.4 Results for the Revert Pattern - Part 2	204
8.5 Activities after Editing	207
8.6 Description of the Labels	207
8.7 Results for the Revert Pattern	208
8.8 Results for the Contemplating/Editing Pattern	209
8.9 Secret Garden	211
8.10 Results for the Creation Pattern	212
8.11 Description of the Labels	213
8.12 Results for the Explaining/Playing Pattern	213
8.13 Results for the Creation Pattern	214
8.14 Results for the Choice Pattern	215
A.1 ERM Example	241
A.2 ERM Diagram of the Knowledge Database	242
A.3 UML Diagram of the Login Classes	243
A.4 UML Diagram of the Management Classes	244
A.5 ERM Diagram for the Revert Tracking Database	245
A.6 UML Diagram for the Revert Tracking Facility	246
A.7 UML of the Behavioural Pattern Description Language Parser	249
A.8 Abstract Class Event	251

List of Tables

2.1	Categories of CSCW Applications	28
4.1	Elements of the Language	83
7.1	Revert Tracking Table	157
7.2	Improved Revert Tracking Table	158
7.3	Adjacency list of a creativity map	161
7.4	Transformation Table	170
7.5	Transformation Table Example	171

Listings

A.1 Pseudo Code for the Identification of a Revert Sequence	247
A.2 Resources Section of the JNLP file	252
A.3 Application Description Section of the JNLP file	252

Chapter 1

Introduction

Objectives

- Discussion of the motivation for introducing behavioural patterns for the analysis of creative processes.
- Explanation of the used research methodology.
- Description of the scope of the thesis.
- Explanation of the structure of the thesis.
- Highlighting of the original contributions.

“If one does not know to which port one is sailing, no wind is favourable.”

Lucius A. Seneca (4 BC - 65 AD)

1.1 Motivation

The first intensive research about the creativity of humans started more than a century ago. Since then, a lot of people have developed interesting models about creativity. Unfortunately, these models cannot explain what makes people creative in the first place. The answer to this question has still to be found. It is an important question. A solution to it would have a big influence to all of our working processes.

There is huge demand from both academia and the economy on approaches, which make people more innovative and not only more productive. Current approaches for a creativity support are still unsatisfactory and cannot cope with the requested demands. The lack of proper approaches is mostly the result of an unclear definition of the term creativity itself. Human creativity seems to be too abstract for a general definition. Therefore, the here presented approach will tackle this problem in a different way through investigating the steps which happened during the creation of the creative product. The thesis will not focus on the analysis the creativity of the creator itself, but his creative process. Through this, it might be possible to learn more about the creative behaviours of the creator and further be able to support him during the creation of an artefact [83].

The presented approach is addressing the creative process as well as the creative product, which distinguishes it from earlier attempts to analyse the creative process and creativity. The creative process is very much influenced by the product. Different tasks cause different creative behaviours. The requirements for a support of creativity differ for each creator. Current approaches are not able to adequately react to those requirements. The here introduced approach maps the creative process in close relation to its effect on the artefact. Through this, it is possible to analyse the effects of the particular activities on the artefact.

Computer technology and especially the internet enable novel approaches for working creatively. The internet provides a vast amount of information in a size which has never been available before. This must have a big impact on creativity nowadays [117]. A second novelty is the ability of working collaboratively, which is enabled through plenty of collaborative technologies. The influence of this on the creative process of the collaborators must be studied as well as the impact of new technologies in general [29].

The in this thesis presented approach describes the storage of the creative process in a special graph, hereinafter referred to as a map. The elements of the map are standardised and mathematically underpinned. This allows the comparison of the creative processes of different creators as well as the processes of different projects of the same creator.

One can see that such a map enables a lot types of analysis. The aim of the thesis is the development of a semi-automatrical analysis framework. The idea is that creators contribute their knowledge about what should be analysed. An information extraction facility as well as the analysis facility will be combined in a tool set. The results of the analysis are finally used for the support of the creative process.

1.2 Research Questions

The motivation for undertaking the research about the creative process is based on a number of open questions. The key question whether it is possible to map and analyse the creative behaviour of creators is quite abstract. It is useful to divide it into a number of subquestions in order to fully understand the scope of the main question.

The questions are:

1. How can information about creative behaviour, which is hidden inside the creativity map, be characterised?
2. Which computer-aided technique enables the extraction of information from creativity maps?
3. Which approach enables the management and store of the knowledge about creative behaviour?
4. What type of information can be extracted from the creativity maps?
5. How can the results of the analyses be used for the support of the creative process?

The research questions emerged in several discussions with scientists from different domains like creative writing, digital music, design and computer science. During these discussions, it became obvious that very similar requirements and demands exist for all domains in regard to the mapping of the human creativity through the capturing of the creative process.

After the development of an initial approach for the capturing and representation of the creative process, it became necessary to analyse the collected data sets in a structured way. The analysis process had to be computer-based due to the huge amount of data. Furthermore, the approach should be flexible and enable the exchange of knowledge. These ideas and their requirements lead to the intensive research, summarised in this thesis.

1.3 Original Contributions

The presented approach for the analysis of the creative process has been made possible through the development of a number of new techniques. A short overview about those techniques, which represent major contributions, is following below.

Behavioural Patterns

Creativity maps contain the information about the creative process of creators. Analysing the creativity requires the identification and extraction of the information about the creative behaviour of the creator. The information about the behaviour is hidden inside the creativity maps. Behavioural patterns are introduced in this thesis as a technique for requesting information about the creative behaviour. The patterns describe the information, which needs to be extracted in order to analyse the creative behaviour.

Behavioural Pattern Description Language

The process of extracting information from the creativity map requires machine-readable descriptions of the behavioural patterns. These descriptions can be expressed through the, for this task developed, Behavioural Pattern Description Language (BPDL). This language is based on the Extended Backus-Naur Form (EBNF). This makes it a flexible and straightforward approach.

Central Knowledge Repository

The analysis of the creative process requires a lot of knowledge in order to achieve the wanted results. A single creator will not be able to gain knowledge about every aspect of analysing the creative process. Hence, there is a demand for a central knowledge database. The in this thesis presented knowledge repository caters for the processing and store of the required knowledge about the creative process. It enables the exchange of behavioural patterns and related information. Its centralised architecture allows workplace-independent access to the knowledge.

Techniques for the Analysis of the Creative Process

The aim of the information extraction process is the analysis of its results. The analysis enables the search for known types of poor creative behaviour as well as the creation of statistics about the create process. The knowledge about the different analysis types is stored in the knowledge repository. However, the process is not only focussing on the

analysis of a single creativity map. It is possible to compare the creative behaviour of two or more creators through the analysis of their creativity maps.

Creativity Support Tool

Some of the results from the analysis process provide more use than simple generation of statistics. A creativity support tool allows the constant monitoring of the creative process. This approach enables the identification of a certain type of behaviour. For example, the identification of poor creative behaviour. Through this, the creativity support tool can be used to inform the creator about the discovery of a particular behaviour at the time of its appearance. Moreover, the analysis process itself will provide a support of creativity as every user will learn more about his and others creative behaviour.

1.4 Scope of Thesis

This thesis contains a number of approaches and technologies, which were developed in order to support the creativity of creators. The focus of the thesis lies on the introduction of a novel approach in the domain of creative technologies. The writing style of the thesis is rather technical than philosophical. The scope of the thesis, therefore, covers mostly technical issues. It mainly includes:

1. Discussion about current approaches and their limitations.
2. Definition of a model for the representation of the creative process with a sound mathematical underpinning.
3. Introduction of the key approaches for the management and store of knowledge as well as the approaches for the analysis and support of the creative process.
4. Presentation of the tool set, which includes:
 - (a) Implementation of a parser for the Behavioural Pattern Description Language.
 - (b) Implementation of the knowledge repository.
 - (c) Implementation of the creativity analysis and support tool.
 - (d) Implementation of the De Montfort Creativity Assistant.
5. Presentation of three case studies for the evaluation of the presented research.

1.5 Research Methodology

The idea of mapping human creativity emerged due to the novel possibilities of modern communication through Web 2.0 technologies and computer technology in general. The project aims for a mathematically sound foundation as well as for a flexibility regarding the field of application. This thesis takes these attributes into account.

The following points provide information about the research methodology, which was used for this thesis. A special focus lies on the collection and analysis of information as it represents the main topic of the here presented work.

1.5.1 Selection of Background Material

Chapter 2 contains material about the background and related work for this thesis. It provides information about the beginnings of the research as well as an introduction of some main researchers on the field with their main contributions. The information in the chapter becomes more and more detailed and recent in publication as it continues.

The material consists mostly of information, provided by conference and journal articles. These contain the most recent research on the topic. On the other side, books provided more general information. Several search engines have been used to find the appropriate content. These include Google Search and Books, CiteSeerX and the online facilities of the De Montfort University Library.

The in this thesis presented research was mostly influenced by the Handbook of Creativity, written by R. Sternberg [120]. It provides a number of very important articles of the most notable scientists in the field of creativity. It represents the ideal source for background information on the field of creativity and computer-supported creative technologies.

1.5.2 Research Method

One of the first and main tasks when undertaking any type of research is the selection of the most suitable research method. This is very important, not only because it is considered as professional good practise, but more importantly because it allows the researcher to coordinate the work. It represents the main working plan for the researcher.

Common Research Methods

There exist numerous research methods for every domain. Therefore, it is not possible to introduce each one in detail. Instead it is more appropriate to introduce the different categories to which the research methods belong. Moreover, there are also different approaches for the categorisation in several domains. In social sciences, the most common model contains two categories [35]:

- Quantitative Research
 - Massive Data Collection
 - Statistical Methods
 - etc.

- Qualitative Research
 - Observation
 - Interviewing
 - etc.

As the name implies, quantitative research methods are used for studies, focussing on the collection of huge data sets and their statistical analysis. The researcher is not involved in the analysed process itself, in order to avoid a contamination of the data. Qualitative research, in contrast, is focussing on interpreting a phenomenon in its natural setting. It often involves case studies, personal experience, interviews and so forth. The researcher himself is often partially involved in the activities [122]. Nonetheless this categorisation is not exclusive. A research method can use elements of both ideas [33].

Research in the domain of social sciences, covering creativity and more specifically the analysis of creativity is often based on the one of these two types of research methods. Chapter 2 contains good examples e.g creativity tests, where these types have been used.

This categorisation is obviously not suitable for research projects with the focus on the development of new approaches and techniques. Therefore, it is not applicable for the majority of research projects in the domain of computer science as neither one of these two categories is important during the creation of something new. The process of developing something novel cannot be adequately described by the quantitative/qualitative research methods. Even if the development process is usually building upon knowledge or information gained through one of the categories.

A common view in computer science, mathematics and natural sciences is that there are three main categories of research methods.

- Exploratory Research
 - Identifying new problems
- Constructive Research
 - Developing solutions for known problems
- Empirical Research
 - Empirical testing of solutions

Exploratory research is used for an identification of new problems within a field [88]. The aim of constructive research is the development of solutions for already identified problems. Research, based on the empirical research method is usually focussing on broad empirical studies in order to verify the usefulness of known solutions on a large scale [69]. One can see that there is a specific order in which these research methods are used within a new field. The list above is deliberately illustrated in this order.

The categories represent the standard procedures of doing research. Specific methods, belonging to one of the categories, derive from these basic methods. However, these methods represent only minimal alterations of the three general categories/methods. The author of this thesis, therefore, used the constructive research as it is appropriate for the demands of the undertaken research.

Constructive research is very often used in computer science. Kasanen describes the constructive research with following words [67]:

“By developing a construction, something that differs profoundly from anything which existed before is created: constructions tend to create new reality.”

He furthermore writes:

“An important characteristic of constructions is that their usability can be demonstrated through implementation of the solution.”

Reasons for the Selection of the Constructive Research Method

The research, described in this thesis, has been undertaken by using the constructive research method. The reason for this choice is the fact that there are already myriads of studies, conducted as exploratory research. Hence, the most important problems in the domain of creativity and creative technologies have been already identified. The conducted literature review provided further confirmation. Missing was a solution for these problems. One of the major research problems was the lack of techniques for the analysis of the creative process (see Chapter 2). As a result, the aim for the undertaken research could only be the development of such a solution. The constructive research method was the obvious choice for this task. Mainly because the research project belongs to the field of computer science and the process of the constructive research is considering the important tasks development, implementation and testing of the new approach.

The steps of the constructive research are described below. Further studies, based on this thesis, might use an empirical method for analysing the data and information gained from the De Montfort Creativity Assistant.

Steps in Constructive Research

Each research method consists of certain steps which the researcher needs to undertake in order to finish the studies successfully. These steps do not necessarily have to be executed in a specific sequence. It is often necessary to adjust the order of the steps to suit the project requirements.

Following is a list with the steps in constructive research [75]. The order of the steps is adjusted for the application on the the research project, described in this thesis.

1. Select a relevant problem with enough research potential.
2. Conduct a thorough literature review.
3. Construct a solution approach.
4. Show the theoretical connections and the research contribution of the approach.
5. Examine the scope of applicability of the approach.
6. Demonstrate that the developed approach works.

Step 1 is represented by the abstract of the thesis as well as by Chapter 1, Section 1.1 - Motivation. The second step is represented by Chapter 2 - Background and Related Work. Steps 3 to 5 are covered in Chapters 3 to 7. The last step, the validation of the approach, is covered by Chapter 8 - Case Studies.

Section 1.3 already provided an overview of the original contributions presented in this thesis. The core of the research was the creation of a framework to generate, apply, store and exchange knowledge about the creative process. These particular tasks required the development of novel techniques in order to provide the required functionality.

A prototype implementation was developed in order to verify the functionality offered by the framework. The framework was tested in a number of case studies as well as used for several other tasks and projects. The results of these case studies as well as a final conclusion can be found in the last two chapters of this thesis.

1.5.3 Data Collection

The data collection process requires the interaction of the user with an application, specially designed for this task. Through this, the user is able to capture his steps when creating an artefact. The provided framework guarantees the storage of the collected data in a proper structure, avoiding an incomplete or corrupt data set.

Mapping the creative process of a creator carries the need for both quantitative and qualitative data input. The collection process takes place alongside the creative work of the user, in order to be as precise as possible, but without hindering the creative process.

Alternative Method

The technique for mapping the creative process allows retracing the creation of an already existing artefact, based on comments and notes of the creator. This enables to analyse the creative process of creators from past decades or even centuries. However, this approach has the disadvantage of possible inaccurate or even wrong analysis results, if the provided information was insufficient. Hence, it can be only an additional technique, in case there is no fresh data available.

1.5.4 Analysis Technique

The analysis of the creative process is based on the knowledge of the creator and people working on similar projects. The analysis process itself is computer-based in order to enable the processing of huge data sets. The used techniques and algorithms for the analysis process are described in detail in this thesis. One of the main objectives for the processing is a flexible approach for users in regard to the information extraction.

The elegance of the provided approach is the fact that the mathematical foundation enables to undertake every possible analysis not only via computer, but also by hand. The here introduced technology represents a very transparent approach.

1.6 Thesis Structure

The following paragraphs provide a description about the content and the structure of this thesis. Every chapter of this thesis is self-contained. The reader can start reading from the beginning or just read a selection of chapters. The content of the chapters are further subdivided into sections and subsections. The non-technical adept readers might start at the beginning for a better understanding of the mathematical models and techniques in the later chapters.

Chapter 1 - Introduction

The chapter Introduction provides the reader with a first overview about the content of the thesis. It discusses the research question, the motivation and original contributions of the writer. The research methodology illustrates the used methods e.g. for the conducting of experiments.

Chapter 2 - Background and Related Research

This chapter contains a literature review on the related work for this thesis. The aim is the clarification of the research gap within the domain of creative technologies and analysis of the human creative process. Furthermore, it provides background information in order to describe the context of this work. The literature review covers the topics creativity, creative process, creativity support and collaboration.

Chapter 3 - Preliminaries

Chapter 3 describes the mathematical underpinning of the in this thesis presented framework. It introduces techniques for mapping the creative process. These enable to describe the activities, used to create an artefact as well as the effect of those on the artefact and the creator itself. The data is used for the creation of a so-called creativity map which depicts the creative behaviour of a creator.

Chapter 4 - Pattern-based Information Extraction

The concept of behavioural patterns and a pattern description language are introduced in Chapter 4. Behavioural patterns allow the description of sequences and segments of creative behaviour. The pattern description language enables represent the behavioural patterns in short expressions, which can be read by a parser. This parser is able to traverse the creativity map. It will search and extract sequences and segments matching a behavioural pattern.

Chapter 5 - Knowledge Management and Store

Behavioural patterns are used to collect the information, used for the analysis of the creative process. The pattern alone will not be of any help without proper knowledge about the usage and purpose of it. This chapter introduces a concept for a knowledge repository, which caters as a central database for all behavioural patterns and their description. The knowledge repository is not only a simple database, but contains a knowledge management facility. This facility organises the adding, accessing and processing of the knowledge.

Chapter 6 - Creativity Analysis and Support

Chapter 6 describes the application of, from the creativity map extracted, information. The behavioural pattern can express the information, which is required for the analysis of the creative process of a creator. The chapter describes different analysis approaches and their use. Finally discussing the support of the creative process through the presented framework. It basically uses the information from the analysis process in order to improve the creativity of the user.

Chapter 7 - Tool Support

Chapter 7 provides detailed information about the tool support for this research project. The first part describes the structure of the web-enabled, collaborative software framework,

the De Montfort Creativity Assistant (DMCA), which was developed for enabling the mapping and analysis of the creative process. The second part describes the techniques, used for the information extraction and analysis process. This includes the algorithms for the creativity map parser. The steps of the parsing process are explained in detail and enriched with examples for a better understanding. The last part of this chapter introduces the two major client side tool-sets of the DMCA: the De Montfort Creative Environment (DMCE) and De Montfort Creativity Mapper (DMCM). Together, both tool-sets enable the support of the creative process of collaborative computer-based projects. The here introduced version of the software provides facilities for creative writers. However, its pluggable design supports the integration of tools for other domains.

Chapter 8 - Case Studies

The chapter Case Studies discusses the validation of the presented approach. It provides application examples with real data. The examples show how the presented research is used for the analysis of the creative behaviour. The case studies were chosen to address and verify the different aspects and functionality of the presented framework.

Chapter 9 - Conclusion and Future Work

The last chapter provides a summary about the thesis and outlines the benefits of the presented research. The usefulness of the chosen research methodology will be discussed. It furthermore contains the conclusion and provides an outlook over possible enhancements.

Chapter 2

Background and Related Work

Objectives

- Introduction of the background of this thesis.
- Description of basic perception of creativity and creative process.
- Discussion of current approaches for the creativity support.
- Description of techniques for the comparison of information and knowledge.

"Invention consists in avoiding the constructing of useless combinations and in constructing the useful combinations which are in infinite minority. To invent is to discern, to choose."

Henri Poincaré (1854-1912)

2.1 Introduction

Before one can start analysing the creative process of a creator with the here introduced technique, it is necessary to have a closer look at the research about creativity and collaboration as well as how new technologies influence these. This chapter provides an overview to recent and past research on the field of creative technologies and point out their limitations. An important point is the description of the term creativity. There exist several models to describe creativity. However, people tend to have different opinions about what creativity is and how it can be enhanced. This chapter will help to understand that creativity is not only a mental process, but influenced by several aspects. It is important to know the common models for the creative process as well as models for creativity itself in order to understand the phenomenon creativity.

The second part of the chapter will discuss the assessment of creativity and the creative process. The role of collaboration for the creative process will be explained in detail. An overview about the requirements for technologies to support creativity as well as current approaches and discussion about their limitations will be provided. A selection of support tools is provided for an overview about the current approaches and technologies for creativity support. Furthermore, an introduction of techniques for the management of information and knowledge is provided. A knowledge database is a key element for any modern creativity support tool. It enables detailed analysis processes and puts the results off those processes into the right context.

2.2 Creativity and the Creative Process

Philosophers and artists have always tried to unveil the secrets behind creativity. The earliest ideas about creativity are from age of the Greek and Roman empires. The philosophers of that time started to think about creativity and genius. It is interesting that the thinking about this topics was much more popular in the western world than in the eastern. Progressive civilisations in the eastern world, mainly Chinese and Indian, shared a different view about creativity. Their view was highly influenced by their cultural and religious believes. Life and creation was understood as a cycle for Buddhists and Taoists - creativity was not associated with novelty, but with modification and adaptation [79]. However, the western view was also not free of religious views. It was believed that creation and creativity lies in the hands of god only. A change of this view was not perceived before the Renaissance. At that time, the thinking changed in a way that great ideas and artefacts were recognised as the achievements of individuals [2].

Researchers started to intensely investigate creativity in the early 20th century. There has been a lot work on creating the foundations of the domain. However, until now there has not been a real breakthrough in unveiling the secrets behind creativity. Many researchers avoided the topic, not without good reason, as it is sometimes considered as a minefield due to the difficulty of evaluating the outcomes of the research.

Margaret Boden wrote following passage in her well received book: “The creative mind: myths and mechanisms”.

“Human creativity is something of a mystery, not to say a paradox. One new idea may be creative, while another is merely new. What is the difference? And how is creativity possible? Creative ideas are unpredictable. Sometimes they even seem to be impossible - and yet they happen. How can that be explained?”[14]

Creativity emerges where existing ideas and resources are formed into new and useful artefacts. It is the combination of common knowledge in a new way [30]. The creative process will form an artefact which represents an object or a concept (novels, music and paintings). The artefact will undergo a (human) judgement whether it fulfils the target (of quality) [108]. Sawyer defined properties to describe everyday creativity. Six facts emerged from his studies of musical and theater performances [114]:

1. Everyday creativity is collaborative.
2. Everyday creativity is improvised.
3. Everyday creativity can't be planned in advance, or carefully revised before execution.
4. Everyday creativity emerges unpredictably from a group of people.
5. Everyday creativity depends on shared cultural knowledge.
6. In everyday creativity, the process is the product.

The listing shows that creativity cannot be understood as a phenomenon, which can be planned or even directly accessed when required. Projects aiming for an on-demand solution of improving the creativity will eventually fail because of the unpredictability of creativity.

Nonetheless, it is possible to study the circumstances, leading to the creation of a creative artefact. Hence, it is necessary to study the meaning of the term creativity and the derived models for enabling a better understanding. Especially the last two points of the listing hint to a way of how the creative process and therefore the creativity of a creator could be influenced and improved. The creative process is playing a major role for the approach, presented in this thesis.

Unfortunately the creative process plays no or only a very minor role in approaches for tools for the support of creativity. As a result, current support tools have remained primitive and unable to cope with the challenges of supporting creativity. The following pages of this chapter will show where current approaches fail and clarify why it is important to analyse the creative process in order to support creativity.

Mel Rhodes had following answer to the question “What is creativity?”:

“The word creativity is a noun naming the phenomenon in which a person communicates a new concept (which is the product). Mental activity (or mental process) is implicit in the definition, and of course no one could conceive of a person living in a vacuum, so the term press is also implicit.”[106]

The question of what is the key to be creative is not easy to answer. Studies indicate that a high intelligence quotient (IQ) will not necessarily lead to a high amount of creativity [71]. There are a handful of abilities which allow people more creative than others.

These abilities are [52]:

- sensitivity to problems
- ideational fluency
- flexibility of set
- ideational novelty
- synthesizing ability
- analysing ability
- reorganizing or redefining ability
- span of ideational structure
- evaluating ability

Nonetheless, every human is capable of achieving creative work to a certain degree. The analysis of differences in the creative processes of people might provide a better understanding of creativity itself.

Sternberg and O'Hara pointed out that there are different opinions for the correlation between IQ and creativity, depending on what aspects of creativity and intelligence are being measured with regard to the domain [121]. Ochse objected that quantifying intelligence is as difficult as quantifying creativity. Hence, it necessary to define the term intelligence before making comparisons about intelligence and creativity [100]. Creativity might be more influenced by the way people think e.g. hemispheric specialisation (preference in usage of the left or right part of the brain), intuition and imagery [10].

Creativity is more and more realised as an important aspect of developing new products [58]. Nowadays companies are searching for solutions to increase the creativity of their staff [82]. This requires the study of the creative process through suitable models. There exist several models for describing the creative process. These models are used for a description of the process in general and therefore cannot provide a detailed analysis. Following is an introduction into some well-known models.

2.2.1 Creative Process Models

One of the first models for the creative process was described by Graham Wallas in 1926. The Wallas model contains four stages of creation [129].

I Preparation

New ideas are always based on already existing knowledge. People usually invent something new on their own field or where they can apply their knowledge. For example scientists need a background of mathematics, biology, etc. to understand and analyse a problem for finally getting new ideas.

II Incubation

Most people will not solve a problem instantly. For many it is stimulating when the task of solving a problem is put aside for a while. Most of us faced already a situation where one got the idea for a problem when making a tea or having a stroll.

III Illumination

Wallas described this as the moment when the new idea appears in the mind. This is the *Eureka!* moment.

IV Verification

It is necessary to examine whether the idea is a useful solution for the problem. The verification is the final step of elaboration and application of the new idea.

The four stages are not entirely separated from each other and can overlap sometimes. It is also possible to visit an early stage again [49]. The Wallas model is sometimes described with having five stages, with an additional third stage called *intimation*. There exist a number of similar models and Wallas' model was not the first. Marchant and Sesko write that it evolved from the Helmholtz Pattern and was influenced by Henri Poincaré. Several other variations followed [85], but never received the same recognition.

Another model for the creative process is called the Directed Creativity Cycle. The concept behind this model is somewhat different to the Wallas model. It is more focussing on the fact that there is usually a repetition in the creative process during the creation of a product. Figure 2.1 depicts the cycle [103].

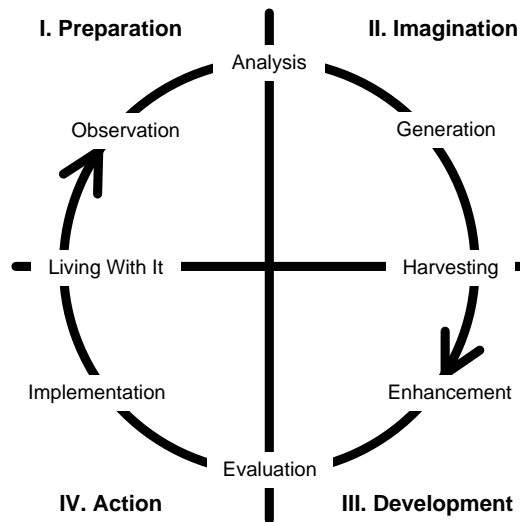


Figure 2.1: The Directed Creativity Cycle

One can see that this model is more fine-grained than the Wallas model. It has similar stages to the Wallas model, but has several sub-stages. Cycles are an often re-occurring element for the description of the creative process. It illustrates that the creative process is not only a task with a start and an end, but with many refinements. The creative process itself will not stop, even the creation of the artefact itself stopped. This can be observed through the fact that creators will usually find some imperfections in their work. The correction of these imperfections would lead to other imperfections and the process would continue.

A closely related topic of the creative process is the creative cognition. Candy describes five elements as part of the creative cognition [24].

These elements are:

- **Ideas generation**

Describes a type of brainstorming, where already existing ideas and methods are combined with each other in order to facilitate the extension and transformation of the design space.

- **Problem formulation**

The identification and formulation of the problems and unanswered questions. The problem space is analysed from different viewpoints to gain a better understanding.

- **Strategies**

This element describes the usage of ambitious or opportunistic steps as a creative strategy.

- **Methods**

The development or acquisition of new methods arises when really necessary. The need for new methods usually occurs due a change of the working strategy.

- **Expert knowledge**

The knowledge of a domain is continually updated through communication over a network of experts of that domain. The type of knowledge is rather informal and recent than documented.

Some researchers doubt that there is a general model for the creative process. In their opinion, the models should be more flexible and fine-grained in order to cope with differences between the domains [80].

2.2.2 Creativity Models

Defining creativity has always been a difficult task. Many models and theories have been developed so far. However, a single precise definition of creativity has still to be found and some call it even a fruitless task [113]. Boden defines two types of creativity, making a distinction between so-called *psychological* and *historical* creativity. P-creativity describes

a situation where a person makes a discovery for himself, regardless how many people came up with it before him. H-creativity is a sub-category of P-creativity, where a person makes a new and unique discovery [14].

Amabile developed a business-related model to describe the important elements for creativity.

These elements are [7]:

- **Expertise**

The knowledge and intellectual property of the company.

- **Creative-thinking skills**

The flexibility of the staff to adapt to new situations as well as their imagination.

- **Motivation**

The motivation of the staff, either by external rewards, like money or commitment to find a solution for a problem.

It was mentioned before that it is very important to create an environment where these three elements are stimulated. This will improve the creativity of the staff [55]. The motivation of the creator plays a key-role for having creative ideas. He must be able to identify himself with the project, in order to feel pleasure and enjoyment for his work. This is necessary for an efficient usage of skills and expertise [54].

An artist or developer requires a convenient working environment for being creative. It is reported that many famous artists knew quite well in what environment they were able to work creatively and productive. Many of them preferred to work in an isolated and quiet environment [6].

The definition of what is considered as creative depends mainly on the society and domain the artefact is received by. The decision whether a product is creative cannot be made by a single person. An Assessment about the creativity of a certain object is only possible through the judgement at the right time and a field of experts, which are highly recognised by others [95, 5]. A product might not be of use or desirable for the majority, but it can still be of value for others. Csikszentmihalyi describes the relation between culture, society and background with his Systems View of Creativity model [34], depicted in the following Figure 2.2.

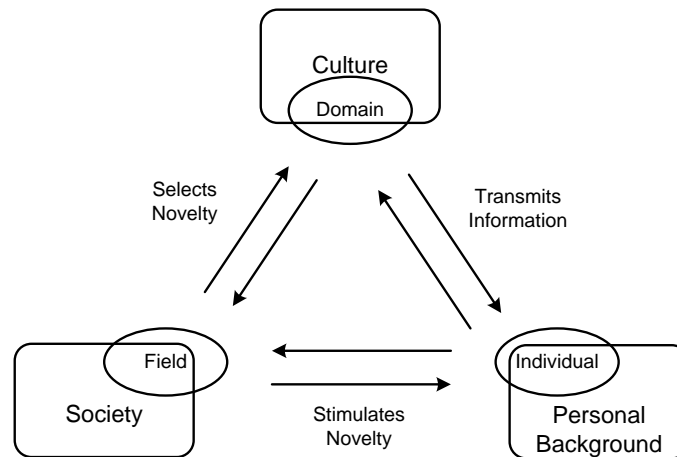


Figure 2.2: The Systems View of Creativity

The model is very useful to understand the complexity of creativity. It is a precise illustration of the relations between the involved parties. The focus lies on the requirements and the stimulation of people, which triggers the need for creativity. It is obvious that creativity has a social and temporal dimension. The socio-cultural environment influences through *what*, by *whom* and *how* creativity is expressed [110].

2.2.3 Creative Thinking Tests

There are a number of tests for creative thinking. One of the simplest methods is the self-assessment, where the creative behaviour of the creator is examined. During the test, the creator is interviewed about his level of creativity. This method is considered not being trustworthy due to the inability of the creator to realise the own creative capabilities [68].

Another widely known method is called Torrance Tests of Creative Thinking (TTCT). It was developed by Ellis P. Torrance in 1966 [125]. The TTCT concept is strongly related to the work of J. Guilford. The test has been revised over the years and several versions of it exist. The test itself measures four factors for identifying the level of creativity. The factors are:

- **Fluency**

The number of interpretable, meaningful, and relevant responses to the stimulus.

- **Flexibility**

The variety of categories of relevant responses.

- **Originality**

The responses which are unexpected, unusual, unique or statistically rare.

- **Elaboration**

The addition of pertinent details.

The test is mostly verbal, but contains some drawing activities too. It measures the quantitative appearances of the mentioned factors. The Torrance test is widely accepted and has been discussed in many research papers. However, it has been criticised as being too abstract for the measurement of general creative behaviour [93].

Another, more recent test was developed by Teresa Amabile in 1982. It is called the Consensual Assessment Technique (CAT) [4]. This test has been developed to avoid the limitations of scoring tests. The scoring tests seem not to assess all attributes of creativity and rather only cover a small area of the creator's creativity. The CAT test is focussing on the creative product and less focussing on the creative process or the creator himself. CAT is well validated in creativity research and therefore often used for various experiments. However, studies indicate that the test is only satisfactory for a small variety of comparisons [9, 57]. These tests are hard if not impossible to integrate into the daily work routine. Moreover it is impossible to cover the whole creation process. The probability of missing the creative moments of the creator is therefore very high. The tests itself are designed for being very short. Another issue is that the tests are always moderated by at least another person. This makes them very expensive.

Thinking Types

Creative thinking tests often distinguish between two types of thinking, known as *Convergent Thinking* and *Divergent Thinking*. Tests which focus on convergent thinking demand for a single solution to a problem, whereas divergent thinking describes the situation of collecting as many solutions as possible to solve a problem [87]. The selection of the more suitable type of test depends on the field of application [53]. The Torrance Tests of Creative Thinking and Consensual Assessment Technique belong to the divergent thinking category.

Overall, creative thinking is closely related to problem-solving processes. There exist many approaches on how people can solve problems. The two major approaches are the finding of a possible solution and the determination whether a solution proposal is suitable. These two categories are referred to as solution-generation and verifying processes [96].

2.3 Collaborative Creativity

Collaboration has always been a source of creativity. People working together bring in different ways of thinking. A group of people will have more knowledge than a single person, especially if people from different domains are involved. In such an environment, experts will use their expertise to provide solutions based on their detailed knowledge for their specific domain.

Creativity often emerges where different ways of looking at a problem are brought together. Analyses have shown that most scientific and artistic innovations emerge from collaboration [47]. Tests with students indicate that the brainstorming and idea-finding part is much more effective when people work in small groups than individually [102]. A single person can only access the knowledge for the own specific domain. Collaborative creativity is understood as people with knowledge of the same or different disciplines, brought together to solve problems, which could be solved without the collaboration of many [25].

2.3.1 Avoidance of Barriers in Creative Collaboration

According to Fischer there are barriers, which might hinder collaboration. It is necessary to identify and overcome these barriers in order to enable collaborative creativity. This is especially of interest when developing software for the support creative collaboration.

Fischer defined the following barriers [46]:

- **spatial**
Managing workplace-independency.
- **temporal**
Dealing with different time zones and asynchronous communication.
- **conceptual**
Reach common ground and shared understanding.
- **technological**
Support of domain-oriented software systems.

The first two points are very important in regard to collaboration. The internet plays a key role for satisfying these two points. Software for the support of collaborative creativity must be web-enabled. Approaches, which are not connected will play a minor role if not completely disappear in future. Hence, these points are very important in an interconnected world.

Bryan-Kinns identified another type of barriers. These barriers are of social nature unlike the just mentioned technical ones. It describes the factors for a satisfactory engagement of every collaborator. This raised three questions of interest [20]:

- **Enticement**

What attracts artists to work on a project?

- **Engagement**

What encourages the artists to collaborate?

- **Sustainment**

What keeps the artists working on the artefact?

Despite the need for an interesting project or task for each collaborator, it is also important that the software itself is attractive to the user. The Graphical User Interface (GUI) of the software should be simple and easy to use in order to avoid hindering the creator when working on the artefact. A lot of tools for the support of creativity fail at this point. Often enough the creativity support is realised through uncommon design of the GUI. But instead of making the collaborator more creative, the software is too confusing as it does not follow the principles of human-computer interaction (HCI) (see Section 2.4.2).

Very important is the question whether the creators are able to work in a team. It requires certain abilities to work collaboratively. The abilities might be [21]:

- **Acknowledgement**

The ability to recognise the contribution of others.

- **Mirroring**

The ability to reflect the contribution of others.

- **Transformation**

The ability to transform and use the contribution of others.

It is necessary that the creativity support tools cater for the reduction of possible tensions [63] between the collaborators and enhance the understanding for the contributions of the others. Technologies like version control help to create awareness of the work of others. Communication facilities support discussions between the collaborators. This enables an easy way to discuss problems or tasks within the team.

The failure of current approaches to cover at least most of the above mentioned points has restricted the field of application. Even more, creative collaboration has rarely been an issue in current approaches so far. As a result, collaboration is a first class citizen for this thesis.

2.3.2 Multimedia Literacy and Transliteracy

For the first time in history, the internet enables a huge number of people to communicate with each other through different types of media. Several multimedia types are used for presentation of content to the viewers of a website. These novel technologies have an effect on the way of how people distribute knowledge and news [37].

The ability to present and consume information through various multimedia types is called multimedia literacy. The term literacy was initially only used to describe the ability to read and write. The online edition of the Encyclopædia Britannica defines the term as the

“...capacity to communicate using inscribed, printed, or electronic signs or symbols for representing language.”

The term multimedia literacy is expanding the meaning of communication by including media formats like audio, text and video. Multimedia becomes more and more important for teaching purposes. Different types of tools have been implemented to provide pupils and students with learnings environments to cater for their need to learn through a wider diversity of media [137]. It is important to note that the focus of multimedia literacy in schools lies not on the media types itself, but the form in which these are used, like narrative, interactive, communicative, adaptive and productive [19].

The interactive fashion of the Web 2.0 and new internet-enabled platforms even enhance the possibilities to communicate beyond those of the multimedia literacy. The term transliteracy is used to describe the ability to use a mixture of platforms and media.

It is defined as

“the ability to read, write and interact across a range of platforms, tools and media from signing and orality through handwriting, print, TV, radio and film, to digital social networks.” [123]

The most important word in this quote is “across”. This term describes today’s technical possibilities to share media not only through one platform, but many different ones. Websites are accessible via standard computers, but also via smartphones and TVs. This has a huge impact on the way people communicate and work. Communication and interaction becomes more and more time and workplace independent. This has definitely enriched people’s creativity and creative process.

As current approaches fail to support collaboration, they certainly fail to cater for a support of multimedia literacy and transliteracy. The De Montfort Creativity Assistant provides several collaboration tools and allows the integration of new tools through its pluggable design.

2.3.3 Computer Supported Cooperative Work and Groupware

Computer Supported Cooperative Work (CSCW) is a multidisciplinary research area, which caters for the understanding of creation, interaction and evaluation of computer supported work. The term was introduced to summarise the research on the upcoming cooperative (or collaborative) work through the usage of computer system during the 1980’s [50]. The main focus of CSCW lies on collaborative applications and their users. One of the main objectives in CSCW is the *awareness* of every involved individual about the collaborative process [38].

CSCW is often mentioned together with the term *groupware*. The two terms are sometimes even referred to as having the identical meaning. However, the term groupware describes the implementation of the principles for CSCW. Groupware is more closely related to product markets whereas CSCW has a more exploratory background [51].

Today most projects in academia and industry are developed and controlled by using computers. Information is stored in central databases, which are world-wide accessible. Members of work-groups are often spread all over the world and might never meet each other. CSCW applications are divided into sub-categories, depending on how the technologies are used.

	Same Place	Different Place
Synchronous	face-to-face group work (electr. meeting room)	sync. distributed group work (video conferencing)
Asynchronous	async. group work (shared computers)	async. distributed group work (e-mail)

Table 2.1: Categories of CSCW Applications

Table 2.1 contains the four different categories of CSCW applications [64]. The categories are chosen by the location of the collaborators as well as the working time. The table itself is referred to as the CSCW matrix. Examples for the categories are provided inside the brackets.

Applications for CSCW require a well designed framework for the user and content management. These applications usually require multi-user support for all four types, described in the above shown matrix. These have a huge effect on the user interface for such an application. It requires the development of flexible communication infrastructures and seamless data management [8].

2.3.4 Computer Supported Collaborative Learning

The Computer Supported Collaborative Learning (CSCL) is closely related to the CSCW. The concepts for both terms emerged during the same time, due to the emergence of personal computers as well as quick and reliable networks.

Today, more and more people use computers for learning purposes. Especially the number of universities which provide online learning environments increased vastly since the last decade [18]. The main objectives of CSCL are the usage of collaborative learning environments with a simultaneous analysis of the progress made by the students [42]. Analyses of CSCL are mostly statistical analyses, based on the content produced by the collaborators [23, 109]. CSCL is used in education for all age groups, starting at the kindergarten [119].

Nonetheless, the CSCL is not only used for analysis purposes, but to introduce new ways of learning. The usage of CSCL can improve the outcomes of the learning process [76]. The reason for this is the potentially availability of huge resources for teaching purposes.

Teachers are able to provide multimedia content, which allows teaching complex ideas to address intellectual challenges [43]. The meaning of the second “C” in CSCL is not precisely defined. Mostly used as the abbreviation for *collaborative*, but sometimes for *cooperative* like in the term CSCW. Some researchers argue that it is necessary to distinct between the two terms. Dillenbourg writes that in collaboration, partners work together on the same problems whereas the work is split when cooperating [36].

2.4 Creativity Support Tools

There is an almost infinite number of classical methods for enhancing the creativity of a person. Every book store has a number of books about this topic in stock. Most people know some of the best known methods like brainstorming, mind maps or Edward de Bono’s Six Thinking Hats. However, due to the technical background of this thesis, only the most common computational methods will be introduced.

Enhancing the creativity of a creator through the support of a computer requires different approaches than the above mention methods. The so-called computational creativity is the attempt to create software which enables the support of creative thinking. It is therefore related to the field of computational intelligence. However, the amount of creative outcome, produced by this approaches, is still very small. One example is a neurocognitive based approach to generate new words and names through combinations of already existing words [39]. In this case, an algorithm is used to find new names for e.g. new products. The words are created through the connection of already existing words, which might describe attributes of the new product.

Other projects use different techniques for example for creative music composition [31] or painting and sculpturing [48]. The majority of the approaches are based upon the same idea of mixing already existing content to create something novel. These projects are facing the problem that they are highly specialised for one domain. There is not a general computational creativity which can cater for all fields. Furthermore, it is doubtful that the software generated output can be considered as creative.

Hence, it is necessary to get a better understanding about what happens during the creation of an artefact and to look behind of what is making people creative. Understanding creativity and the creative process is a difficult task. Therefore, some might ask how it is even possible to enhance the creativity of a person through computer technology.

Hewett emphasizes in his works that:

“We may not yet understand enough about the conditions under which creativity is going to happen to be able to help it along, but we do understand some of the things that make it harder for creativity to happen. Knowing what disrupts creativity makes it possible to figure out ways of staying out of the way and not interfering.” [56]

The term Creativity Support Tools (CST) covers a broad variety of software. Some researchers use the term for software, which will allow users to develop creative products. However, this could be said for many applications. A novelist could write a fantastic book with Microsoft Word. Hence, all these software applications are not actively involved into enhancing the creativity of the user. All creativity is initiated by the user himself. In a widely recognised paper, Shneiderman discussed applications and websites like Google Search, Wikipedia or MatLab and Flash as tools to support creativity. However, he also recognised CST as actively influencing applications.

“... A second example of creativity support is the capacity for users to rapidly generate multiple alternatives, explore their implications, or revert to earlier stages when needed” [116]

The support of creativity always requires the knowledge about design processes. A system for the store and exchange of knowledge must satisfy following features [26].

- Knowledge modification and evolution.

Accessible and powerful methods for modifying the knowledge base should be provided so that the users can refine existing rules and add new rules incrementally at any time during the on-going tasks.

- Evaluation of evolving knowledge.

Support for evaluation of the evolving knowledge in progress should be provided. The user must be able to ask why or why not about the results of any request for evaluation and receive an explanation that is expressed in domain specific terms.

- Comparative evaluation of knowledge.

Support for comparative evaluation of the knowledge in the system should be available. The user should be able to create and evaluate selected rule sets with different variables against identical criteria allowing immediate comparisons of their performance relative to each other and to a standard rule base.

Most of today's software to support creativity lacks of the above mentioned principles. The focus of current creativity tools lies on visual stimulation and not on the processes behind creativity. These approaches often try to stimulate the creator through a special design of the GUI. The idea is that a different concept for the working process can provide more stimuli to enhance the creativity of the user. The majority of researchers questions the effectiveness of those concepts for following reasons:

- The creator will get used to the environment.
- People might not like a change of the working environment.
- The approach will not have the same effect on all creators and might even hinder the creativity.

2.4.1 Impact of New Technologies on Creativity

Computers and other digital media have changed the way people live today. Creators have access to almost unlimited possibilities of making creative art through digital media and software applications [40]. The internet provides unlimited access to knowledge and information. Furthermore, communication between people became much more convenient. Technologies, like email, instant messaging, but also mobile phones make it easier to contact others. Before, a letter took several days from the sender to the recipient. Emails are now sent and received within seconds.

A problem of the information age is the growing amount of data and the resulting complexity of managing it. People more and more often lose the overview about their own data. The efficient management of information plays a key-role for the success of CST. Computer technology can and does support people to find the information they need. Furthermore, technology support enhances the creative potential by providing new information at the right time [44].

2.4.2 Human-Computer Interaction

The domain of human-computer interaction (HCI) is very wide defined and therefore divided into several sub-domains. It is basically the study of the interaction of humans with computers. This includes input and output devices as well as the design of user interfaces of software applications.

The majority of CST is based on HCI design principles. The goal is the development of software applications which enable the user to be more productive through the usage of a sophisticated graphical user interface. The developer of a software application needs to follow certain design criteria in order to satisfy the requirements of HCI. The following list contains a number of criteria, which are mandatory for development process of software for the support of creativity [105].

- **Support of Exploration**

Possibility to try many different alternatives.

- **Low Threshold**

Good software should always be self-explaining and only require a short settling in period.

- **Support of Variability**

The user can choose different ways to achieve the same aim.

- **Support of Collaboration**

Provide groupware functionality.

- **Support of Open Interchange**

Seamless integration of different tools.

- **Simplicity**

Avoiding complex user interfaces through good usability.

- **Careful Choosing of Black Boxes**

Determine which ideas users can explore with the tool - and which ideas remain hidden.

- **Invent Things That You Would Want To Use Yourself**

An application is only useful, if the developer himself enjoys using it himself.

- **Balance of User Suggestions**

Choose a right balance of user involvement.

- **Prototyping**

Creation of many prototypes for a constant improvement.

- **Creative Design**

Provide creative stimulus for the user.

- **Evaluation**

Evaluation is necessary, even it is difficult to achieve.

It was already mentioned that HCI represents the research in a certain number of sub-domains about the usage of computers and similar platforms. In recent years, user interface design has moved from a very strict menu structure to a more flexible mixture of menus and buttons. An example for this is the current version of Microsoft Office, where a new toolbar provides common functions.

The principles of human-computer interaction are very important for the development of Creativity Support Tools. Without these principles, new tools might not empower the creative user. Lately, a growing number of researches have been undertaken in order to understand the nature of creative cognition and to enable the evaluation of CST [27].

2.4.3 Examples of Creativity Support Tools

The presented tools represent the state-of-the-art technology of this domain. Most of the available applications are developed by small companies rather than having emerged in an academic environment. The following list provides a selection of well-known CST.

AXON Idea Processor

The AXON Idea Processor is designed to support the thinking process of its users. The application is implemented in Prolog and therefore follows the concept of using facts and rules to provide answers to a certain question.

The application provides a GUI which enables the user to create complex flowcharts and diagrams. The main focus of AXON lies on the support of brainstorming sessions [15].

The current version provides instant messaging facilities in order to enrich the system with communication and collaboration functionalities.

IdeaFisher/ThoughtOffice

IdeaFisher is a commercial tool set for the support of lateral thinking and brainstorming sessions. It uses a database with words organised by categories. Based on this database, it supports the user to create “mental associations” for a word or a concept. The support itself is realised through cross-referenced database entries.

The creativity support is realised through the generation of new ideas based on word combinations. It automatically records the results on a so-called Idea Notepad. The latest releases of the software are distributed under the name ThoughtOffice [104].

freesound radio

Freesound radio is a collaborative tool for the exploration of music compositions. The tool enables the creation of new music samples, which can be shared with others. The project offers a repository with audio samples, licensed under Creative Commons.

The software itself is fully web-based, providing a simple GUI. The creativity support is realised through mixing sounds and music in a collaborative environment. The results are stored in a central databank and analysed for similarities [94].

RiTa

The RiTa toolkit was created for experiments with generative literature in the domain of creative writing. It covers a number of computational tasks like text analysis, text generation, text mining as well as access to external resources like ontologies. The text-generation is based on Markov chains and grammars [61]. The analyses mainly focus on the structure and composition of text.

2.5 Limitations of Current Creativity Support Tools

Although the research on Creativity Support Tools has created many ideas on how to support the creator with his work, it still lacks of a personalised creativity support. The concept of most support tools is based on the theory that key for more creativity lies in

the usage of different tool-sets. Furthermore, the methods still build on visual stimulus for providing a creative environment. The applications are very often limited for the usage in one domain only. Ben Shneiderman presented his technology genex, which is based upon following aims [115]:

- Collect information from an existing domain of knowledge.
- Create innovations using advanced tools.
- Consult with peers or mentors in the field.
- Disseminate the results widely.

All these points are important for the support of creativity. However, there is no personalised creativity support mentioned.

The common understanding of CST is that these act as a platform for managing an artefact as well as providing knowledge and information for a certain domain. Lubart proposed that computers should facilitate four different tasks for the support of creativity [81]:

- The management of creative work.
- Communication between individuals collaborating on creative projects.
- The use of creativity enhancement techniques.
- The creative act through integrated human-computer cooperation during idea production.

This categorisation also does not mention an analysis of the activities made during the creation of an artefact. The lack of studies on the creative process leads to a huge research gap in the domain of creative technologies.

There have been only a few attempts for mapping the creative process of creators. One example: Participants of a workshop were asked to complete a task with the help of computers. The whole process was filmed in order to record the activities of the user. The activities were later examined and listed for each participant [66]. This system, however, lacks of an automatical analysis and provides only a shallow reflection of the creative process.

The limitations of the discussed technologies as well as the criteria for the development of an efficient tool set for the analysis and support of the creative processes have been taken into account for the framework, introduced in this thesis.

2.6 Knowledge Management

The research on creativity and the creative process is closely connected to the field of knowledge management and knowledge engineering. The analysis of creativity always requires and also produces new knowledge. Hence, it is necessary to introduce a knowledge management facility for the process of analysing creativity and the creative process. The knowledge is usually managed via specially designed frameworks. These knowledge management frameworks play an important role in the domain of computer science and many computer-based projects. A project community can use this type of framework to learn from, contribute to and collectively increase and improve the knowledge of the community [12].

Knowledge Management, however, is not only required for analysis purposes, but necessary as a foundation and support for the creation of novel products in industry and academia. The research on knowledge management emerged in large corporations. The main aim was the preservation of the employees' knowledge in order to avoid a critical loss in case of large staff fluctuation or downsizing [101]. Knowledge is sometimes considered to be the most important asset of every company [16, 128].

The knowledge stored inside a knowledge management facility is subject to the following life cycle [111]:

1. Originate/create knowledge
2. Capture/acquire knowledge
3. Transform/organise knowledge
4. Deploy/access knowledge
5. Apply knowledge

These five steps cover all, with the knowledge management facility related, activities. Any software application, designed as a knowledge management facility, must support these actions.

The understanding of knowledge management has changed over the years. It started somewhat as a hype, but the topic received less attention in the last couple of years. In both academia and industry, it is considered more and more as not the only important or the central instance [41], but one of many important pieces of the creation process [1].

2.6.1 Data Preparation and Retrieval

The exchange of knowledge and information via modern communication devices bears certain difficulties. A large knowledge database must provide search facilities in order to enable a fast and easy access. The search facilities require a tagging of the knowledge for a categorisation with so-called metadata. The in the internet provided information is heavily depending on the usage of metadata. It simplifies the work of website crawler. The Hypertext Markup Language (HTML) even contains language elements for the handling of keywords for a website.

The content of the knowledge database is seldomly created by a single person. As a result, there will be differences in the labelling and lead to the problem of using different terms carrying the same meaning. This problem requires techniques catering for the identification of the right context of words.

Polysemy and Synonymy of Words

Polysemy and synonymy are serious problems in the domain of natural language processing. The fact that many words in a language have more than one meaning causes problems for comparisons. A good example is the word “bank”, which can be a seating-accommodation or a financial institution. As a result, people looking for a financial institution with the aid of a search engine could receive information about a nice place in a nearby park. The phenomenon of multiple meanings for a word is known as polysemy.

The other, more important issue in regard to tagging information in databases is called synonymy. Each language has sets of words, which have the same or a similar meaning. An example are the words “thinking” and “contemplating”. Both words have a very similar meaning and it is very useful for searches to also include the entries, which are tagged with terms, similar to the original term of the search query. A lot of research has been undertaken so far in order to tackle these problems.

The problems with the ambiguity of words has led to the emergence of a new research area in computer science, called word sense disambiguation (WSD). The aim of WSD is the development of techniques, which enable to identify the meaning of words in a given text. This aspect plays an important role in text mining and information retrieval processes [112]. Hence, it is worth to take notice of WSD as it uses the same techniques for the identification process, which are being used to handle metadata in knowledge databases. The first approaches in WSD were made in 1950's with more or less poor techniques [62]. One technique used specialised databases which contain terms for a specific, minimal topic. The data sets were often manually enriched with metadata. The disadvantage of this approach is quite obvious as it is impractical for texts with versatile topics [127]. Furthermore, the method is very time consuming.

The research about WSD quickly became connected to the domain of artificial intelligence (AI). The development of the AI approaches began in the 1960's. Improved approaches tried to solve the language understanding with a type of rule-based system. The disambiguation of the data was realised through the introduction of large knowledge databases for semantics and syntax of a language. The systems then produced semantic representations of the text in the data set. Some of the systems already utilised the relations between nouns and verbs. Furthermore, some of these methods used ontologies or case-based approaches. The rules for the WSD process were quite simple and less precise than the approaches in the following decades. Especially the synonymy and polysemy were difficult to handle which those approaches.

The following paragraphs introduce the most common techniques for solving the issue of polysemy and synonymy. The discussed techniques are useful in regard to the tagging of data and information in knowledge databases as well as WSD for text.

Knowledge-based Approach

The aim of the knowledge-based technique is the usage of machine-readable dictionaries [134]. The approach reaches a high precision on a defined topic area. Hence, the systems are quite poor at handling text with a different context. There are three different types of knowledge sources: machine-readable dictionaries, machine-readable thesauri and machine-readable lexicons. The knowledge-base approaches were much later invented than the AI techniques because of the lack of knowledge databases before the 1980's. The further development of these techniques lead to the corpus-based approach.

Corpus-based Approach

The corpus-based approach is based on dictionaries and thesauri, enriched with information, which were gained from statistics over large corpora [78]. The field of research is divided into two different categories, supervised techniques and unsupervised techniques. Supervised techniques are very precise in the disambiguation of words. This is caused by the human trimmed database. Unfortunately, the supervised method is not applicable for projects, which cater for a large variety in topics. The unsupervised techniques often use machine-readable dictionaries or thesauri for the preprocessing of the text. Yarowski developed an interesting method for unsupervised WSD. In his approach, he assumes that an ambiguous word just occurs with one sense of it in the majority of texts. Furthermore, the approach considers the collocations in regard to the corpora. Collocations are sequences of terms which often occur together. For instance, a noun is usually used with certain set of verbs. It assumes that there is one sense per collocation. This means that nearby words help to describe the meaning of the ambiguous word. Knowing this, the algorithm collects seed collocations for an incremental process for the identification of collocations for target senses. His approach is almost as powerful as supervised techniques [135].

WordNet

WordNet is a sophisticated thesaurus for the usage of computer based language processing. The project is focussing on the development of a semantic lexicon which surpasses the functionality of a common thesaurus. It is being developed at the Princeton University and is currently available in the 3rd version (June, 2010) [91]. The WordNet project is designed for the English language. However, there are already similar projects for other languages. One of them is the PersiaNet, the Persian WordNet [70].

The software and the database of the project is available for download from the project homepage (<http://wordnet.princeton.edu>). The software runs under Windows and various UNIX and Linux operating systems. Furthermore, APIs for different programming languages are existent, for instance a Java API or Prolog version. The software license is similar to GPL and allows changing the source code and a usage of the software for commercial applications. Furthermore, an online service of WordNet is available as well.

The database of WordNet contains information about nouns, verbs, adjectives and adverbs, so called "open-class words". Other types like determiners, prepositions, pronouns, conjunctions, and particles are left out of consideration. The semantic database is a hybrid of a dictionary and a thesaurus. Nouns and verbs are categorised into hierarchies.

Hence, it has knowledge about the affiliation of a word. A simplified example would be: oak \Rightarrow tree \Rightarrow plant. Another very useful feature is the knowledge about synonymy and polysemy [32].

Self-Organizing Map

A Self-Organizing Map (SOM), also known as Kohonen Map, is a clustering technique, closely related to artificial neural networks. This technique converts the nonlinear statistical relationships between high-dimensional data into low-dimensional maps. It utilises an unsupervised learning approach for the visualisation of high-dimensional input data without losing the most important relationships of the primary data elements [74].

SOMs are used for various analyses, pattern recognition and categorisation tasks. The technique requires a training with the data set in order to generate the map. During the training or learning step, an array of nodes (neurons), each one containing a weight vector, is arranged by the weight of the vector. The training is repeated for many times (up to several thousand) until no significant changes are observed anymore [73].

Self-Organizing Maps enable a good categorisation of terms, based on the relation to each other. It is a good technique for the identification of similar terms in order to manage e.g. information or documents. A project, called WEBSOM was initiated for the exploration of the information provided by the internet, by enabling interactive browsing [59]. The WEBSOM technique has proven to be able to handle massive data sets and is therefore capable of being used for large projects [77].

The drawback of the SOM approach is the need of a preprocessing and training if new elements are added to the data set.

2.6.2 Conclusion

Knowledge management in general becomes more and more important. Knowledge is considered as being the key asset of a company. It is necessary to focus on an efficient distribution and storage in collaborative environments. Without a proper knowledge management, creativity and innovation might be hindered severely.

Data preparation is a necessary part for the management of knowledge in collaborative environments. The requirement of preparation increases with the variety of information stored in the data set. Especially frameworks with large a diversity of users like Web 2.0 compliant systems should remove the ambiguity of terms as good as possible [90].

2.7 Summary

This chapter contains a summary of the major research in the domains of creativity, creativity support and knowledge management. The material was selected in order to provide a background for the in this thesis presented research. It would be impossible to cover every aspect of these fields due to the huge amount of information created through the research and studies which were undertaken so far. Nonetheless, the information should assist the reader and point out the research gap, which this thesis aims to eliminate.

The first pages of this chapter inform the reader about the beginning of the research about creativity. The literature review has shown that there are still a lot of questions that need to be answered in order to fully understand human creativity. It also shows the requirement of new approaches for identifying the nature of creativity. A very promising starting point is the creative process. The creative artefact is the sum of all actions undertaken during the creative process of one or many creators.

Several models for the description of creativity and the creative process have been discussed. An important point is the knowledge that not the creator himself is judging the creativity of the artefact, but the society which receives it. Obviously, creativity is not an isolated phenomenon and therefore involves more than the efforts a single person.

The creative process models show the major stages of the creation process. Cycles are an important attribute of the creative process and will be therefore an re-occurring element in this thesis.

Collaborative creativity is another field with many unexplored properties. Modern technologies, mainly the internet and web-enabled applications boost the possibilities of collaboration. Before the world-wide web, it was difficult to manage time and workplace independency. Communication and collaboration in real-time was difficult and expensive. The research on related work has shown the importance of collaborative software for helping workgroups to improve the communication and exchange of knowledge.

The chapter provides a detailed introduction into the requirements of creativity support tools. Furthermore, examples of current creativity support tools are provided. A discussion of the limitations of those approaches explains why there has not been a breakthrough yet and what has been missing so far in each of the approaches.

Another disadvantage of current creativity support tools is the absence of a knowledge repository. Without this, it is impossible to store knowledge about the creative process of

a creator. This makes it impossible to follow the personal development of the creator. For instance, it is impossible to find out whether the creativity of the creator has improved when using the creativity support tool. Another drawback is the fact that creators in a collaborative environment can't exchange their knowledge properly and in a structured way.

The development of a knowledge repository for a collaborative environment is not straight forward. Each person has a different vocabulary for the description of events and artefacts. This problem is clearly recognisable when people from different domains work together. A mathematician will describe his environment with different words than a painter. Hence, this chapter introduces and discusses several methods as a solution to this problem.

Chapter 3

Preliminaries

Objectives

- Presentation of a novel approach to map the creative process.
- Introduction of the mathematical underpinning for the presented approach.
- Introduction of creativity maps.
- Explanation of a hiding operator for the refinement of creativity maps.

“Don’t worry about people stealing your ideas. If your ideas are any good, you’ll have to ram them down people’s throats.”

Howard H. Aiken (1900-1973)

3.1 Introduction

This chapter provides the introduction into the mathematical framework, which is used as the foundation for the storage and analysis of the creative process of creators. A generalised transition system caters for the mapping of the creativity of creators. The first section will explain the idea of mapping human creativity and how it can be achieved. The following sections will show in detail how the creative process and therefore the creativity is mapped by a generalised transition system. The chapter provides a description of the entities for representing the work flow of the creator. The last section before the summary will discuss the creation of a so-called creativity map from the collected raw-data. Techniques for the partial hiding of information will be presented as well.

3.2 Mapping Human Creativity

The presented approach for analysing the human creativity is based on a generalised transition system. The idea of mapping human creativity is the possibility to follow the creator through every single step he or she made when creating a product. This is a novel and advanced technique in regard to the usual approaches of investigating the creative process through psychological perception. The steps either describe mental activities in the creator's mind e.g. *contemplating*, *inspiration* or physical activities like *reading* or *writing*. It is possible to store this behaviour by recording the activities of the creator and the effects of these on the artefact. The collected information enables a better understanding of what happens during the creative phase of a person. It enables the extraction of the collected data for a detailed analysis.

Tracking the steps of a creative person requires a versatile model with the ability to store and represent the collected data. The model should enable a detailed analysis and be mathematically sound. Flexibility is very important as the model has to be domain independent and therefore capable of mapping the creativity of, for example, a musician as well as of a computer scientist. Furthermore, every artist or scientist has different working methods which need to be covered. Without this flexibility, it will not be possible to map all activities and situations the creator might go through. Another important attribute is simplicity. The presented approach is flexible and capable of describing complex structures, but simple enough to allow to reproduce the results on paper. The model and its mathematical underpinning are based upon standard structures from the domain of theoretical computer science. The chapter contains references to standard works in the domain of automata theory, which provide background information for the gentle reader.

The following sections of this chapter will introduce the idea of how a generalised transition system is used for mapping human creativity.

3.2.1 Axioms of Creativity

Following is a definition of axioms for creativity. Three premises are formulated, which are believed to be fundamental to understand and hence to analyse the notion of creativity [136]. The axioms can be seen as a refinement to the already available models about creativity and the creative process, with a special focus on mapping human creativity.

(A0) Creativity is identified only by its product.

Creativity can only be analysed if the resulting product is present for an analysis. For example, the genius of a painter can only be witnessed if one can have a look at his paintings. Creativity always flows into the product. Creativity can only be identified if the product can be reviewed, even it is as small and simple as a formula. There is no possibility for a proof of creativity without a product.

(A1) The value of creativity is determined only by the society which receives it.

A phenomenon of creativity is that the question, whether a product is creative or not, is depending on the society which receives it. In art for example, the works of an artist often become popular a long time after their creation. Sometimes even after the death of the artist. Therefore, it is obvious that the determination of creativity for a certain type of art can change. A different community might interpret a product more creative than others. As a result, it highly depends on the people and their reaction to a certain product. Different communities use different metrics for the evaluation. The metrics will define what is highly creative and what is not creative at all. Naturally, the metrics can change or be replaced by others.

(A2) Creativity is an emergent phenomenon.

Creativity is a spontaneous and chaotic phenomenon. Artists and scientists often refer to the *Eureka! event* when being asked to provide a description about their creativity. It is impossible to plan for creativity as it a mostly subconscious process. The result of creative behaviour is more than the summary of its influencing parts.

It is impossible to foresee the event of emergent creativity. A new state will usually appear unexpected. Therefore, one can only analyse the event a posteriori. New states appear on a non-regular basis. These states can either be desirable or be chaotic. It is not possible to distinguish between these due to the emergent behaviour.

Creativity as an emergent phenomenon is environment-dependent. A state which is chaotic in one environment might be desirable in another environment.

The axioms are used to define the characteristics of creativity. These axioms provide a summary of the most accepted models for creativity like the Systems View of Creativity by Csikszentmihalyi presented in Chapter 2.

3.3 Generalised Transition System

The recording of the creative process of creators requires a flexible model which can represent the collected data in a structured way. It is necessary to understand the basic elements of the creative process in order to identify the best choice. Let us therefore analyse quickly what a creator is doing when working creatively. A creator undertakes hundreds or thousands of activities when working. The activities or actions lead the creator from one stage of the process to the next one. In computer science, these actions are described as *transitions* and the stages as *states* in a *transition system* [92]. An advantage of the transition system is that it can be easily depicted as a directed graph. This is very helpful as these are easier to understand than the notation of formulas.

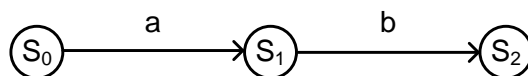


Figure 3.1: A Transition System

Transition systems belong to the field of automata theory and are similar to the finite state automata, but with possible infinite set of states and transitions. Transitions describe actions which lead from one state to another. Multiple transitions between two states are not possible. Therefore, traditional transition systems are not capable of coping with the more-dimensional creative process. As a result a generalised transition system (GTS) was introduced [136].

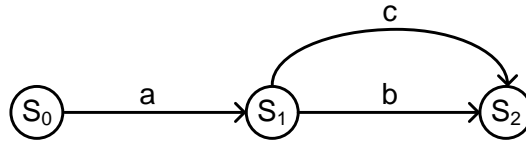


Figure 3.2: A Generalised Transition System

Let the generalised transition system be a tuple (Σ, T) containing a set of states Σ and a set of transitions T . Then \rightarrow_t is the relation between two states, $t \in \Sigma \times \Sigma$.

3.4 States

One of the most important facts for every analysis is that the more detailed data is available the better and more detailed the results will become. Any progress made by a creator will let him reach a new stage on the way to the completion of his work. In the here presented model, states store the changes or updates made to the artefact by the creator if available. A state contains the summary of past behaviour [17].

A state in this generalised transition system is never empty. It contains a vector with different types of data. The amount of data inside the vector can differ and is depending to the domain it is used for. One element however is always present: a timestamp. The timestamp is required for the positioning of the state in a time line. Another element of the vector is usually the current version of the product itself at time the state was created.

$$S = \langle E_1, E_2, E_3, \dots \rangle$$

The vector elements of the state S . In finite state machines, this type of states are often referred to as “fat” states [13].

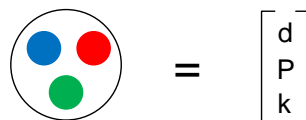


Figure 3.3: State

Figure 3.3 provides a graphical representation of a state with its content. The content of it is represented as a vector on the right of it. The first inner circle for example could stand for the timestamp d , the second circle for a version of the product P and the third circle for knowledge k , gained during the process by the creator.

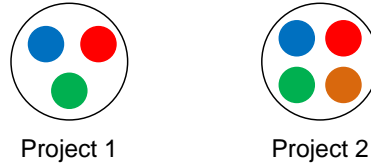


Figure 3.4: States of different Projects

Different projects can require different elements in the state vectors. It is also possible that two states of the same project contain different elements. An example would be a tag at a specific state, created by the creator.

3.5 Transitions

Creators always have a set of activities which they apply during the creation of a product. These activities are repeatedly used, where one activity is followed by another. The activities are modelled as transitions inside the generalised transition system.

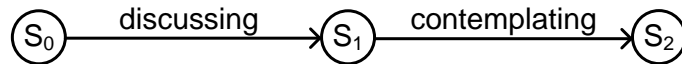


Figure 3.5: Transitions between States

A transition in the generalised transition system is defined by four properties. These four properties are used to describe which relation it defines, what type activity is represented by it and what viewpoint it belongs to.

$$t = (S_0, S_1, L, V)$$

Every transition t requires an initial state S_0 and the state it is leading to S_1 . The variable L describes the label of the activity, which creates the new state. The last variable V

describes the viewpoint of the transition, which is not of interest for now and will be explained in section Viewpoints underneath.

The set of labels for the transitions is defined by the users of the system. Every creator can personalise his set of activities for each project, thus it will fit perfectly for the description of the creative behaviour.

3.5.1 Activity Types

The activities during the creation of a piece of art belong to different categories. Some activities will influence the state of a product in a different way than others. The majority of the activities will not directly change the product, but might influence the mind of the creator. Only the minority of transitions will directly change the product itself. As a result, two different types of activities can be defined. Let us call them Creational Activities and Intellectual Activities.

Intellectual Activities

The majority of activities are of intellectual nature. These are the activities which do not directly change the product. Following is a list of examples for intellectual activities.

- Contemplating
- Reading
- Discussing
- Recovering
- etc.

When looking at this list of activities, one can see that these will influence the mind or intellect of the creator. A product or artefact will not be change directly through discussions or contemplating. The intellectual activities are acting passively towards the process of creation.

Creational Activities

The other category contains activities, which directly change the product. These activities are therefore called creational activities. Examples for creational activities are:

- Writing
- Painting
- Composing
- etc.

It is necessary to distinguish between these two categories. The intellectual activities are a type of preparation for the creational activities. A creator will not be able to write a sentence or make a brushstroke without prior thinking about it. Hence, the intellectual activities are highly influencing the creational activities.

In some cases, the categorisation of the activities is a difficult task. For example a novelist might make notes during his work. This would be an intellectual activity as it will not change the product itself, but will help the novelist memorising an idea. Therefore making notes and editing the novel must be defined as two separate activities inside the generalised transition system. Otherwise, a later analysis of these activities will produce false results. The creator himself is responsible for the correct categorisation of the activities. This is necessary to avoid producing faulty results during the analysis through the above mentioned problem.

3.5.2 Start and End of a Transition

It is useful to provide a formal definition for the start and end of a transition. As every transition has an initial state from where it is directing to a final state, following functions are valid.

The *start* (St) of a transition is defined for a given transition $\sigma_i r \sigma_j$ as:

$$St : \Sigma \times \Sigma \longrightarrow \Sigma .$$

if $\sigma_i r \sigma_j$ then $St(r) = \sigma_i$.

The *end* (Et) of a transition is defined for a given transition $\sigma_i r \sigma_j$ as:

$$Et : \Sigma \times \Sigma \longrightarrow \Sigma .$$

if $\sigma_i r \sigma_j$ then $Et(r) = \sigma_j$.

In the generalised transition system, a state can have several leaving transitions and several incoming transitions. However, in the later introduced creativity maps, each state has only one incoming transition.

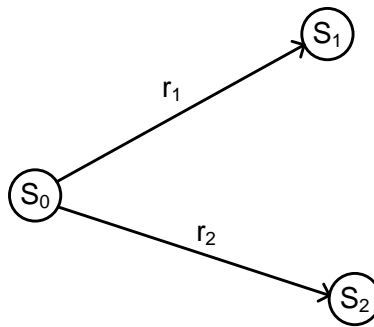


Figure 3.6: Transitions and States

The example of a GTS in Figure 3.6 has following properties:

$$St(r_1) = St(r_2) = S_0,$$

$$Et(r_1) = S_1 \text{ and } Et(r_2) = S_2$$

The start and end of a transition enable the definition of its direction within the generalised transition system. This enables the analysis of the flow of all transitions.

$$St(R) \hat{=} \bigcup_{r \in R} St(r) \text{ and } Et(R) \hat{=} \bigcup_{r \in R} Et(r)$$

3.5.3 Scope of Influence

It was mentioned before that the state does not represent a single object, but a vector containing several objects. A transition, leading from the current state to a new state, does

not change all elements of the new state's vector. The through the transition represented action can only change one element. For example a *writing* transition will only change the artefact and not the knowledge of the creator. A *contemplating* transition however will only change the knowledge and not the artefact itself.

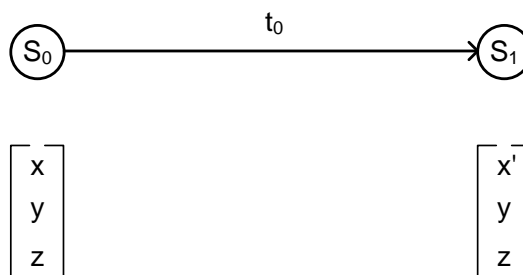


Figure 3.7: Scope of influence

Figure 3.7 depicts two states S_0 and S_1 , connected through a transition t_0 . The two states contain a vector with three elements each. One can see in the figure, that transition t_0 only changed the first element x into x' . The other two elements were not touched by the transition.

Obviously, the artefact can only be changed through the creational activities of the creativity map. The change of a knowledge element, if existing, depends on an intellectual activity, like *contemplating*, *reading* or *discussing*. One element of the vector, the timestamp, is not changed by an activity. It is set automatically and therefore does not apply to this behaviour.

3.5.4 Special Transitions

There exist some special transition types in the generalised transition system. The special transitions are necessary for a complete formal description of the generalised transition system.

ϵ -Transition

The ϵ -transition only applies if the creator finished working on the project. In this case the ϵ -transition acts as a wildcard until a possible continuation of the project in future.

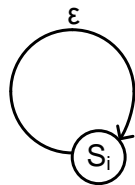


Figure 3.8: ϵ -transition

Figure 3.8 depicts an ϵ -transition. It is the only transition where the initial state equals the end state.

An ϵ -transition is defined as:

$$\forall s \in \Sigma : s \epsilon s$$

for any Σ .

ANY-Transition

The *ANY*-transition represents the set of all possible transitions. It is defined as:

$$ANY \hat{=} \Sigma \times \Sigma$$

for any Σ . The *ANY*-transition is also called the *maximum* transition.

NULL-transition

A *NULL*-transition ϕ is defined as:

$$\forall \sigma_1, \sigma_2 \in \Sigma : \neg(\sigma_1 \phi \sigma_2),$$

for any Σ , where $\neg(\longrightarrow_r) \hat{=} (\Sigma \times \Sigma) \setminus \longrightarrow_r$. The *NULL*-transition is the minimal relation. This transition is not existing within the space of possible transitions.

3.6 Operations

This section describes the operations that can be applied within the generalised transition system. These operations are necessary to represent the creative process and therefore the creative behaviour as a creativity map.

3.6.1 Choices

Choices are substantial for mapping the human creativity in a generalised transition system. The following three types of choices, namely general choice, branching and merging, play a vital role for the creation of a map of the creator's creative behaviour and the succeeding analysis.

Mapping human creativity would not be possible without being able to describe choices. The creative process is never linear. Creators often go back to a previous idea, after failing to continue with the current one. In this case, the creator will further improve the previous idea by following a different approach, instead of going the same alley again, which led to the situation of being stuck. This type of event is represented by a branch in transition systems, like the creativity map.

General Choice

It is necessary to introduce the operator, describing the general choice, before introducing an operator for the branching itself. The general choice operator enables the expression of a choice of two or more transitions for connecting two states.

The general choice operator $+$ is defined as:

$$\longrightarrow_{r_1} + \longrightarrow_{r_2} \hat{=} \{(\sigma_i, \sigma_j) : (\sigma_i \longrightarrow_{r_1} \sigma_j) \vee (\sigma_i \longrightarrow_{r_2} \sigma_j)\}$$

The two states (σ_i, σ_j) in the example above can be either connected through transition r_1 or r_2 .

Branching

Branching occurs where a creator went back to a previous idea and created a new creative path, beginning from that particular state (idea).

The branching operator $\tilde{+}$ is defined as:

$$\longrightarrow_{r_1} \tilde{+} \longrightarrow_{r_2} \hat{=} \begin{cases} \longrightarrow_{r_1} + \longrightarrow_{r_2}, & St(\longrightarrow_{r_1}) = St(\longrightarrow_{r_2}) \wedge \\ & Et(\longrightarrow_{r_1}) \neq Et(\longrightarrow_{r_2}) \\ \phi & otherwise \end{cases}$$

A branch is created if two or more transitions have the same initial state, but point to different states.

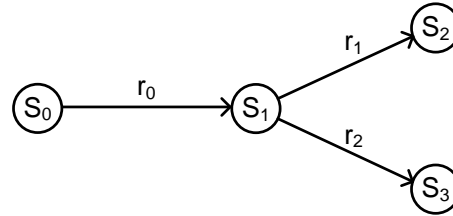


Figure 3.9: Branching

The example in Figure 3.9 shows the branching in a transition system. It contains two transitions (r_1, r_2) which lead to two different states from the same initial state.

Merging

The merging and branching operation are structured similarly. In case of a merging, two states are pointing to one state. This happens in case of reverting to an older state.

The merging operation \pm is defined as:

$$\longrightarrow_{r_1} \pm \longrightarrow_{r_2} \hat{=} \begin{cases} \longrightarrow_{r_1} + \longrightarrow_{r_2}, & Et(\longrightarrow_{r_1}) = Et(\longrightarrow_{r_2}) \wedge \\ & St(\longrightarrow_{r_1}) \neq St(\longrightarrow_{r_2}) \\ \phi & otherwise \end{cases}$$

In words, this expression describes two states as merging into one state, if the ends of both transitions point to the same state and both transitions have different initial states.

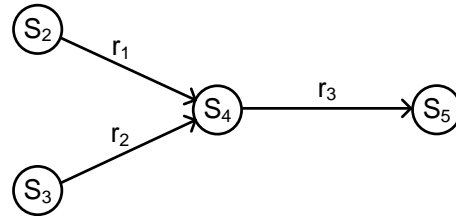


Figure 3.10: Merging

Figure 3.10 depicts the graphical representation of a merging operation. One can see two transitions (r_1, r_2) pointing to the same state.

3.7 Sequences

It is impossible to perform a detailed analysis based on a single transition. Through this represented information is not sufficient to make any predictions on the influence of this one transition in regard to the whole creation process. Therefore, it is more important to be able to analyse a sequence of transitions. These sequences are so-called creative paths. A creative path represents the steps of the creator's creative behaviour. This allows to retrace the steps the creator went through during the creation of an artefact. This data already enables several types of analysing the creative behaviour of the creator.

3.7.1 Sequential Composition

The sequential operator ($;$) enables to describe the composition of transitions. It is now possible e.g. to describe a composition of two transitions r_i and r_j for any Σ as:

$$\sigma_1 \xrightarrow{r_i} \sigma_2 ; \sigma_3 \xrightarrow{r_j} \sigma_4 \hat{=} \begin{cases} \sigma_1 \longrightarrow \sigma \longrightarrow \sigma_4, & \sigma = \sigma_2 = \sigma_3 \\ \phi, & \textit{otherwise} \end{cases}$$

The composition of several transitions represents a segment of the *creative path* of a creator.

3.7.2 Creative Path

A creative path \rightarrow^* describes the sequence of actions or activities, performed by a creator, in a linear way. The sequence can consist of zero or more transitions for any Σ .

$$\rightarrow^* \hat{=} \epsilon + \rightarrow ; \rightarrow^*$$

The linear composition of the creative path does not enable to depict branching and merging operations. A creative path is only a graphical representation of the various activities, recorded by the De Montfort Creativity Mapper. The transitions in a creative path follow a timeline and are therefore arranged by their timestamp. The entities of a creative path are not more than raw data.

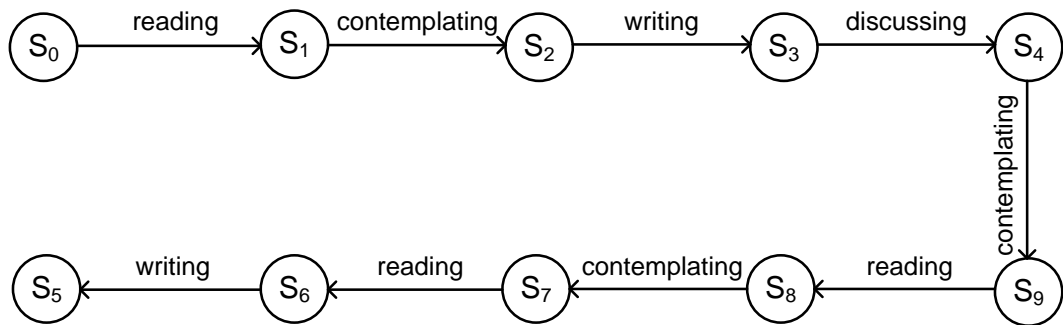


Figure 3.11: A Creative Path

Figure 3.11 depicts a typical creative path. The labels of the transitions indicate the various activities of the creator when working on the artefact. As already mentioned, the clock progressed with every transition (activity). This means that the creator started reading, then contemplating about what he or she had read and then wrote down the idea and so forth.

Every creative path will be different as every creator has an own individual working process. However, different paths might often contain similar parts of behaviour. Either within the same path or when compared with others. This would be very interesting for an analysis. Nonetheless, the creative path cannot display the complete creative process and is therefore not suitable for the mapping of the creative process. It is only the basis for the creation of a creativity map, which will be discussed later in this chapter.

3.8 Laws of Creative Processes

The transition system is based on algebraic rules in order to enable its correct creation and furthermore the manipulation of the collected data. The presented laws define the behaviour of the choice (+) and sequence (;) operators [60].

3.8.1 Identity and Units

$$\begin{aligned} \phi + \longrightarrow_r &= \longrightarrow_r = \longrightarrow_r + \phi \\ \epsilon ; \longrightarrow_r &= \longrightarrow_r = \longrightarrow_r ; \epsilon \end{aligned}$$

3.8.2 Zeros

$$\begin{aligned} ANY + \longrightarrow_r &= ANY = \longrightarrow_r + ANY \\ \phi ; \longrightarrow_r &= \phi = \longrightarrow_r ; \phi \text{ (finite)} \end{aligned}$$

3.8.3 Commutativity

The choice (+) operator is commutative.

$$\longrightarrow_r + \longrightarrow_s = \longrightarrow_s + \longrightarrow_r$$

The sequence (;) operator is not, because the order of the transitions matters.

$$\longrightarrow_r ; \longrightarrow_s \neq \longrightarrow_s ; \longrightarrow_r$$

3.8.4 Associativity

Both the choice and the sequence operator are associative.

$$\longrightarrow_r + (\longrightarrow_s + \longrightarrow_t) = (\longrightarrow_r + \longrightarrow_s) + \longrightarrow_t$$

$$\longrightarrow_r ; (\longrightarrow_s ; \longrightarrow_t) = (\longrightarrow_r ; \longrightarrow_s) ; \longrightarrow_t$$

3.8.5 Distributivity

Same situation as for the Associativity. Both the choice and the sequence operator are distributive.

$$\longrightarrow_r ; (\longrightarrow_s + \longrightarrow_t) = (\longrightarrow_r ; \longrightarrow_s) + (\longrightarrow_r ; \longrightarrow_t)$$

3.8.6 Star and General Rules

Following, the rules which apply for the generalised transition system. The star (*) notation indicates zero or more appearances of the operand.

$$(I) \phi^* = \epsilon$$

$$(II) \epsilon^* = \epsilon$$

$$(III) ANY^* = T \quad , \text{with } T = (\Sigma \times \Sigma)^*$$

$$(IV) \longrightarrow_r^* = \epsilon + \longrightarrow_r ; \longrightarrow_r^* \\ = \epsilon + \longrightarrow_r^* ; \longrightarrow_r$$

$$(V) (\longrightarrow_s + \longrightarrow_r)^* = \longrightarrow_s^* ; (\longrightarrow_r ; \longrightarrow_s^*)^*$$

$$(VI) \longrightarrow_{r_1} + \longrightarrow_{r_1} = \longrightarrow_{r_1}$$

The first five points describe the effect of the repetition operator (*). The last point describes the rule for the choice between the same transitions.

3.9 Viewpoints

Activities are not only categorised into the already mentioned intellectual and creational types. It is useful to further categorise them into so-called viewpoints. By its definition, a transition can only change one entry of the state vector, e.g the artefact itself. On the other hand, the entry of a state vector can be changed by more than one transition type (activity).

Assuming that the knowledge entry (K) of the state vector could be changed by a reading transition ($R = \langle \textit{reading}, \longrightarrow_{\textit{reading}} \rangle$) and a contemplating transition ($C = \langle \textit{contemplating}, \longrightarrow_{\textit{contemplating}} \rangle$).

The two transition types R, C would belong to viewpoint knowledge K :

$$K = R \cup C.$$

In other words, a viewpoint represents all transition types which change the same entry of the state vector.

Viewpoints allow to focus on specific activities during the analysis and allow to hide unwanted information. For the one analysis, it might just be of interest to analyse transitions which affected the knowledge of the creator. Another analysis could focus on creational transitions. Viewpoints are important for addressing the different dimensions of the creative process. This technique enables a flexible analysis. Through this, it possible to analyse e.g. the influence of two types of transitions on each other.

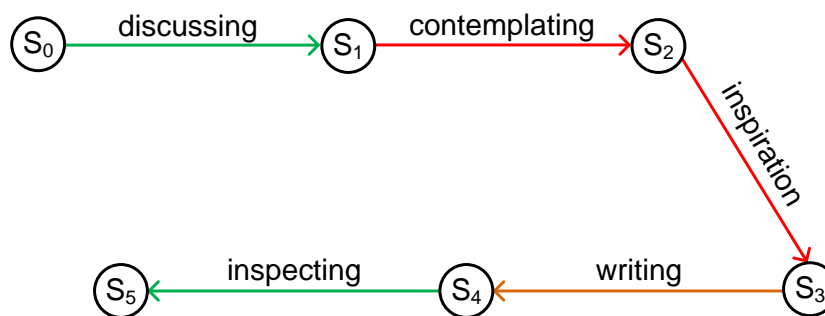


Figure 3.12: Colour-coded Transitions

Viewpoints are expressed within the creativity map through **colour-coded** transitions. This improves the awareness of the particular viewpoints and the, with those viewpoints linked, activities. There are five transitions in Figure 3.12. The colours indicate that there are three different viewpoints. These could be for example called **Need**, **Knowledge** and **Creation**.

The **Need** (N) category contains the discussing and inspecting activities.

$$D = \langle \text{discussing}, \longrightarrow_{\text{discussing}} \rangle \text{ and } I = \langle \text{inspecting}, \longrightarrow_{\text{inspecting}} \rangle$$

$$N = \text{discussing} \cup \text{inspecting} \text{ is the state-space of the } \textit{Need} \text{ category.}$$

Next is the **Knowledge** (K) category, which contains the contemplating and inspiration activities.

$$C = \langle \textit{contemplating}, \longrightarrow_{\textit{contemplating}} \rangle \text{ and } \textit{Ins} = \langle \textit{inspiration}, \longrightarrow_{\textit{inspiration}} \rangle$$

$$K = \textit{contemplating} \cup \textit{inspiration} \text{ is the state-space of the Knowledge category.}$$

Last category is **Creation** (Cr), containing only the writing activity.

$$W = \langle \textit{writing}, \longrightarrow_{\textit{writing}} \rangle$$

$$\textit{Cr} = \textit{writing} \text{ is the state-space of the Creation category.}$$

An activity can only belong to one viewpoint. The set of activities would require further refinement if one activity belongs to more than one viewpoint. In this case, the set of activities usually needs to be subdivided into a more fine-grained level. For example, if *writing* belongs to the viewpoints *Creation* and *Knowledge*, it is possible to split the activity into *editing* (*Creation*) and *making notes* (*Knowledge*). This will avoid flaws in the result set of a later analysis.

NB: Please be aware of the fact that the colours for each viewpoint are assigned randomly. Therefore, the colours can (and usually do!) change between different creativity maps!

3.10 Creativity Maps

The creative path represents the collected raw data from the creator. The creative process of the creator is not visible at this point, but still hidden inside the creative path. The information about the creative process needs to be extracted before it can be analysed. This, however, requires a new model for the representation of the creative process, as it is not linear, but complex and chaotic in its structure.

The solution to this problem is the so-called creativity map. It enables to represent the creative process as a whole. Branching and merging operations are used to map those behaviours, where the creator got stuck with his current approach and went back to a previous stage of the project and continued working from that stage using a different approach.

3.10.1 Semantics of the Creativity Map

The structure of the creative path is unfortunately not reflecting the creative behaviour of the creator, but only the chronological order of the transitions. The information about the creative process is hidden inside the creative path and needs to be extracted before any analysis can take place.

In a creative path, the *start* of the new transition t_{n+1} is always the *end* of the previous (last) transition t_n in the set of all transitions T :

$$St(t_{n+1}) = Et(t_n),$$

if $t_n \in T$ and $T \neq \emptyset$.

One can see that the creative path is a simple concatenation of transitions. This representation can be used to map parts of the creative process. A complete mapping of the creative process is not possible, as the creator will get stuck once in a while and try other approaches. This kind of behaviour is not covered by this representation model.

The transitions in a creativity map are not arranged by their time of appearance, but their **logical** position within the creative process. The *start* of a new transition can be the *end* of any transition t_x :

$$St(t_{n+1}) = Et(t_x),$$

with $t_x \in T$ and $T \neq \emptyset$.

The *start* of transition t_{n+1} is either the *end* of the last transition t_n or, and this is the important difference in regard to the creative path, the *end* of another transition in the set of transitions ($t_x \in T$), which has been chosen by the creator himself.

The latter case is triggered by a revert activity, where the creator went back to a previous state. This state can e.g. be chosen because the creator had created a version of the artefact, which he or she considers as better than the current one. Another meaning could be that the state represents an important idea, which the creator wants to further develop by taking it into a different direction. In either way, the revert activity always means that the creator got stuck with his approach and therefore needed to go back to a previous state and continued from there.

The revert operation leads to the creation of branches in the creativity map. Through this, it is possible to map all aspects of the creative process. Its structure represents the logical image of the creative process. Unlike the creative path, the creativity map has to be generated. The generation process transforms the creative path into a creativity map, using the information about the revert operations.

Revert Activities

Keeping track of the revert activities is very important for mapping human creativity. Without this information, it would be impossible to analyse why a creator got stuck with an idea and more importantly how he or she solved the problem.

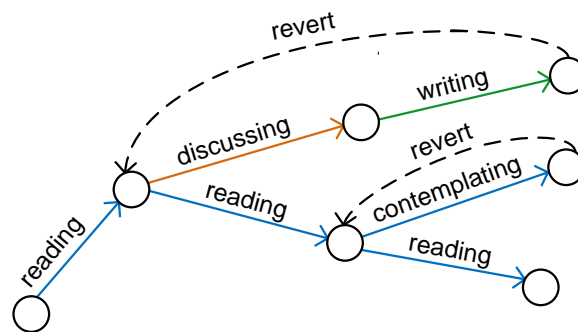


Figure 3.13: A Revert Action

Figure 3.13 enables a better understanding of the *revert* operation, which is usually hidden inside the creativity map. The revert operation is depicted through a dashed arrow.

Theory and practice of reverting to a previous state differ a little bit. For example, the creator would not lose the knowledge he or she gained before returning to a previous state of the project. When switching back to an old idea, the creator would select an old version of the artefact. This could be e.g. a previous version of a document. However, a revert operation is not limited to states, which were created through changes to the artefact. It can also point to a state, representing the growth of knowledge or an inspiration.

The information about the revert operations is provided by the creator, because only the creator can decide to go back to a previous version. The process for the creation of the creativity map depends on this information.

The next figure depicts the actual creativity map. The information about the revert events is hidden in order to maintain the simple structure of the creativity map.

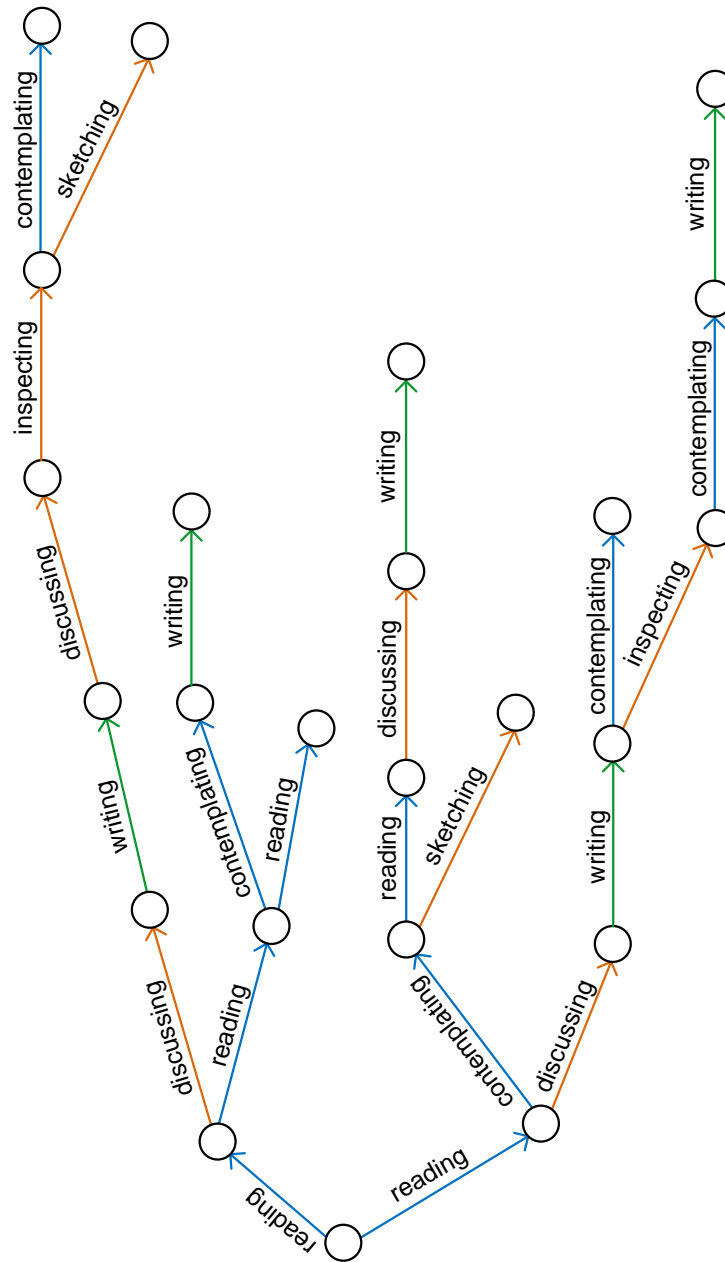


Figure 3.14: A Creativity Map

Figure 3.14 depicts a creativity map with various transitions and a couple of branches. The

example is the creativity map of a creative writer. The branches indicate that the creator went back to an old idea and started a new creative path from it on several occasions. Please note that the branches do not always start with a state, which was created by a *writing* transition.

The example contains three viewpoints:

Creation with $\langle \longrightarrow_{writing} \rangle$,

Need with $\langle \longrightarrow_{discussing}, \longrightarrow_{inspecting}, \longrightarrow_{sketching} \rangle$

and **Knowledge** with $\langle \longrightarrow_{reading}, \longrightarrow_{contemplating} \rangle$

The three viewpoints show a typical distribution of activities. The viewpoint of the creational activities, here *Creation*, is usually the one with the smallest number of activities. Other viewpoints, describing intellectual activities, in this example *Need* and *Knowledge*, can have a lot more activities, depending on the type of project and domain.

3.10.2 Partial Creativity Maps

Partial creativity maps (PCM) represent special creativity maps, where certain information of the original creativity map has been hidden. The hiding can be based on one or more viewpoints or one or more elements of the state vector. The idea behind the PCM is derived of the fact that some analyses of the creative behaviour require a reduced set of information. For instance, if one is interested in the correlation of two viewpoints. A postprocessing of the analysis results would be complex and could even lead to false results. Hence, it is more convenient to hide the unwanted information in advance.

The partial creativity map simplifies the extraction of information for a subset of states and transitions. The hiding process facilitates a reduction of the complexity of a creativity map. Partial creativity maps can be created, through two different ways: *viewpoint-based* and *state-based* hiding. The first type enables the analysis of the correlation between specific viewpoints. This is realised through hiding the transitions of unwanted viewpoints in the creativity map. The second type of partial creativity maps is based on certain entries inside the state vector. A transition only indicates that an entry of the state vector has been changed. The transition cannot provide any information about how the change has affected the entry itself. This is a problem for the analysis, because the creator is not only interested to know *what* entry has been changed, but also *how* the entry has changed.

This type of information can only be extracted from the state itself. A PCM of this type will support analyses, which e.g. only focus on parts of the creativity map, where a large growth of the artefact can be observed.

A partial creativity map is derived from a normal creativity map and shares the same attributes and operations. Partial creativity maps can be automatically generated [130]. The later in this thesis introduced analysis approaches are applicable for the normal creativity maps as well as partial creativity maps.

Viewpoint-based Hiding

Partial creativity maps, which are based on viewpoints, represent a subset of information about the creator's creative process.

A normal creativity map, as a creative structure C [136], can be described as:

$$\langle C, \rightarrow_C \rangle = \langle T, \rightarrow_T \rangle \times \bigcup_i \langle A_i, \rightarrow_{A_i} \rangle$$

The viewpoint T represents the temporal dimension of the creative process. It forms the Cartesian product with the set all viewpoints $A = \bigcup_i A_i$, which are of interest.

The partial creativity map can be described as a creative structure P with following attributes:

$$\langle P, \rightarrow_P \rangle = \langle T, \rightarrow_T \rangle \times \left(\bigcup_i \langle A_i, \rightarrow_{A_i} \rangle \setminus \bigcup_j \langle H_j, \rightarrow_{H_j} \rangle \right)$$

The partial creativity map is a structure, where only a reduced set of viewpoints is of interest ($A \setminus H$ with $A = \bigcup_i A_i$ and $H = \bigcup_j H_j$). Again, the set A represents all viewpoints, except viewpoint time, and set H the viewpoints which will be hidden.

The viewpoint time T has a special status as each activity will take a certain amount time for its execution. This information cannot be excluded from any creative structure. Hence, it is excluded from the set of viewpoints which are of interest for an analysis. It can never belong to the set of hidden viewpoints.

The information about the viewpoint V is stored as one of the properties of a transition t .

$$t = (S_1, S_2, L, \mathbf{V})$$

The entry is crucial for the creation of viewpoint-based PCMs. The decision, whether a transition is taken into account for the creation, is made by analysing this entry. This is the reason why a transition can only belong to one viewpoint.

It is important to note that the start and end states of transitions inside a viewpoint-based PCM might be different compared to the original creativity map and other PCMs.

Two examples, based on the creativity map in Figure 3.14 will help to illustrate the nature of partial creativity map (PCM)s. The first example, a partial creativity map with only the transitions of the viewpoint *Knowledge*, would require the following set of hidden viewpoints H :

$$H = \langle \textit{Creation}, \longrightarrow_{\textit{Creation}} \rangle \cup \langle \textit{Need}, \longrightarrow_{\textit{Need}} \rangle$$

The resulting PCM is depicted below.

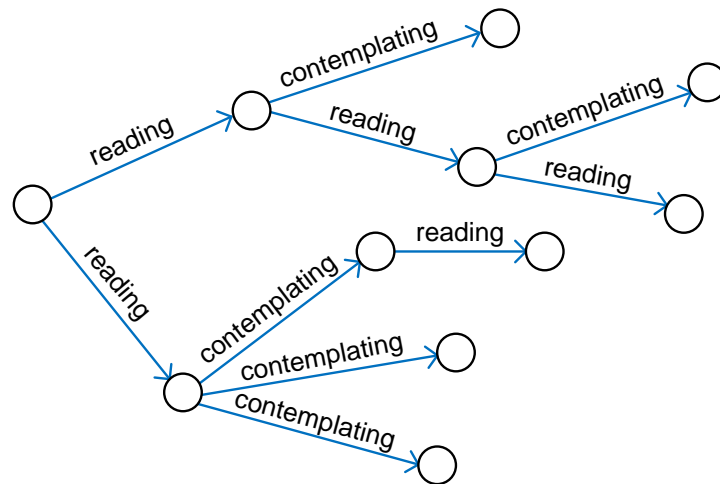


Figure 3.15: Partial Creativity Map based on one Viewpoint

It is still a creativity map with different types of activities. However the blue colour of the transitions indicates that these activities all belong to the same viewpoint. Obviously

a partial creativity map, consisting of one viewpoint, is useful for analysing the influence of a set of activities on the product.

One might now be interested in the correlation of e.g. the transitions of the viewpoints *Creation* and *Knowledge*. In this case, the viewpoint *Creation* will be removed from the set of hidden viewpoints H :

$$H = \langle \textit{Need}, \longrightarrow_{\textit{Need}} \rangle$$

Only the *Need* viewpoint remains in the set of hidden viewpoints. This will result into a larger creativity map, containing more information.

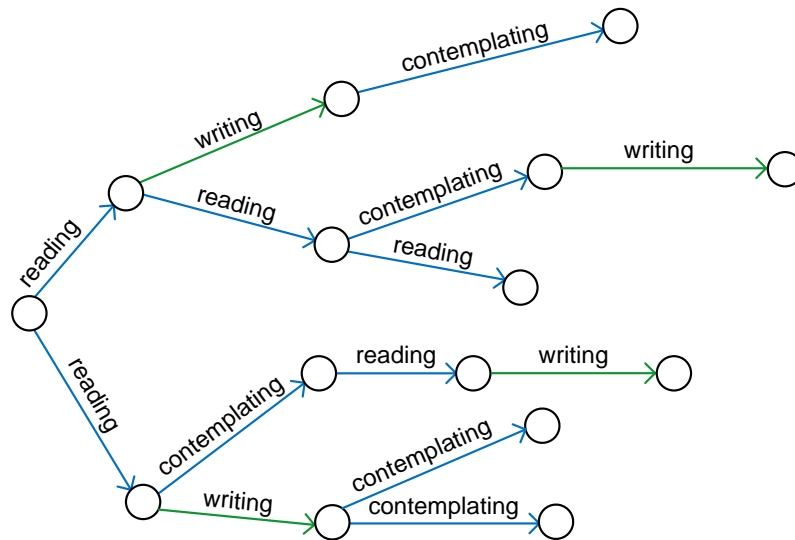


Figure 3.16: Partial Creativity Map based on two Viewpoints

Figure 3.16 depicts the partial creativity map with the two viewpoints *Creation* and *Knowledge*. One can see the additional *Creation* viewpoint in form of the green *writing* transitions. The order of the transitions is equivalent to the order in the original creativity map. With other words: The position of the transitions has not changed. Only the transitions of unwanted viewpoints are hidden in the partial creativity map.

State-based Hiding

The creation of a state-based PCM is similar to the viewpoint-based approach. This type of PCM hides information, depending on one or more entries inside the state vector. The entry of the state vector can only be changed by the transitions of the associated viewpoint. Furthermore, the same entry is usually not changed consecutively. Transitions, belonging to different viewpoints will alternate during the creative process.

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} \xrightarrow{a} \begin{bmatrix} x' \\ y \\ z \end{bmatrix} \xrightarrow{b} \begin{bmatrix} x' \\ y' \\ z \end{bmatrix} \xrightarrow{a} \begin{bmatrix} x'' \\ y' \\ z \end{bmatrix} \xrightarrow{c} \begin{bmatrix} x'' \\ y' \\ z' \end{bmatrix}$$

Figure 3.17: Alternating Transitions

In figure 3.17 it is shown how the entries of the state vector are changed by the transitions \xrightarrow{a} , \xrightarrow{b} , \xrightarrow{c} . Each transition belongs to a different viewpoint and, therefore, changes a different entry of the state vector.

The entry z of the state vector could for example represent the artefact. Often, someone might to analyse what happened between to changes and simultaneously wants to reduce the amount of information which needs to be analysed. Hence, it is useful to define a certain *criteria* by which the information is filtered, so that only information remains which promises to be meaningful. This kind of information is obviously not stored inside the transitions. It is necessary to access the states to gain this information.

The following example will illustrate this approach.

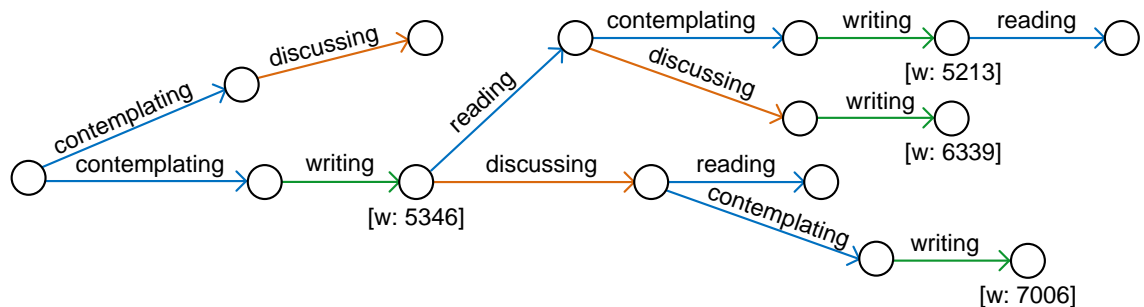


Figure 3.18: Creativity Map Example

Figure 3.18 depicts a creativity map example. It is the creativity map of a creative writer, working on a novel. It is assumed that the creator in this example wants to improve his ability of writing large texts in one piece, instead of writing lots of small parts. Therefore, it is necessary to create a PCM, only consisting of creative paths which have lead to a large increase of the word count. Creative paths with only a minor increase or even a decrease of the word count are hidden. The word count of the document is denoted below the states in the figure. The number is only shown directly after the artefact has been changed. Nonetheless, the information about the word count is stored in each state of this creativity map.

In this example, it is assumed that only an increase of the word count by more than a thousand words is considered as a good performance. It is also the criteria by which the parts of the original creativity map will be hidden.

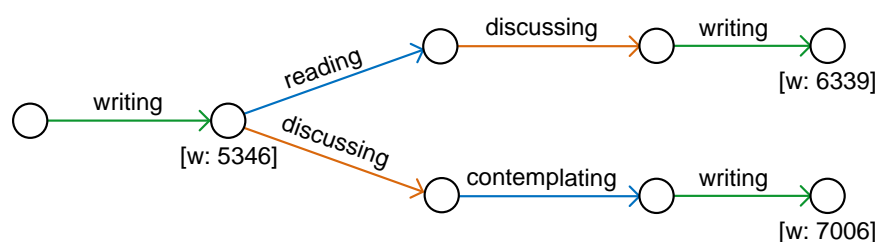


Figure 3.19: Partial Creativity Map

Figure 3.19 depicts the resulting partial creativity map. The amount of states and transitions has been reduced drastically. One can see that the PCM contains only paths with a state at the end, where a large increase of the word count could be observed. One state, where the artefact had also been changed, has not been taken into account as the condition was not satisfied (word count decreased). Other parts of the creativity map, where there have been no changes to the artefact, have been hidden as well.

The state-based PCMs seem to overlap with transition-based PCMs. However, this is not the case. States inside the creativity map can contain user-defined tags. These tags cannot be addressed by the viewpoint-based approach. The state-based approach also enables a much more fine-grained view at the attributes of the entries in a state vector.

Further Reading

The topic of this thesis is not the creation of creativity maps and partial creativity maps, but the analysis of these. The objective of this chapter is the introduction of the models and what these represent. A complete description of all aspects of the creation process for the creativity maps and especially the partial creativity maps would go beyond the scope of this thesis. More detailed information is available in another thesis [130], which is focussing on these topics.

3.10.3 Uninfluenceable Activities

Uninfluenceable activities belong to a category of transitions which contain only a few activities which constantly occur during the creation of an artefact. These special transitions usually run parallel to all other transitions in the creativity map. This type of transitions is not helpful for most of the analysis. At the same time, the information can be stored inside the states. The information itself is not lost. As a result, these transitions are usually hidden in the creativity map, to reduce the amount of data. However, these special transitions shall not be unmentioned for the sake of completeness.

The following transitions belong to this category:

Time Transition

The time transition marks the time spent on the creation of an artefact. This transition runs parallel to every normal transition. The information of this transition is somewhat trivial and usually not required for the analysis.

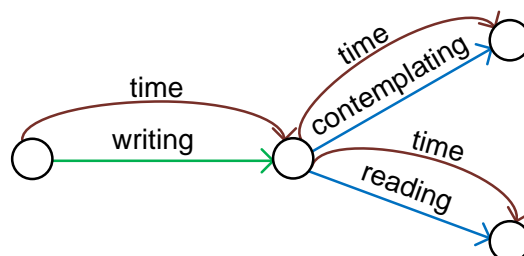


Figure 3.20: Part of a Creativity Map with Parallel Transitions

Figure 3.20 depicts a part of a creativity map with time transition. One can imagine that these transitions are quite confusing in a larger creativity map. Furthermore, every state contains the actual timestamp. The time transition only indicates that the clock progressed.

Thinking Transition

Every person is constantly thinking during the whole life, even when sleeping. It is impossible for a human to stop subconscious thinking. People think when reading, writing and so on. Therefore, there could be a thinking transition parallel to every normal transition - same as the time transition. However, these thinking transitions are not important for a later analysis. Only an action, resulting from the subconscious process, like inspiration, is of interest.

3.11 Summary

This chapter provides the formal underpinning for the process of mapping the human creativity. The generalised transition system (GTS) was introduced as a flexible model. It represents the ideal foundation for the analysis of the creative process. The, on the GTS based, creativity maps are capable of representing the creative process in its entirety. The precision and granularity of the creativity map is only limited by the collection method of the raw data and the resulting accurateness.

The GTS consists of two elements: states and transitions. States represent the different stages or versions of the creative product. The content of a state is not limited to the artefact itself, but several elements, stored in a state vector. Those elements can be for example the artefact, gained knowledge and quality aspects (like the results from discussions or inspections of the artefact).

Actions are represented as transitions from one state to a second one. The end state of the transition is usually a new state, but can also be an old state in special cases. Furthermore, all transitions are labelled, describing the current action. Through this, it is possible to describe the whole creative process, including the creational activity but also intellectual activities like contemplating, reading, discussion and so on. This thesis is focussing on the activities and their representation as transitions in the creativity map. The activities are the atomic elements of the creative process.

Each activity belongs to a certain viewpoint. A viewpoint represents all activities which change the same element of the state vector. For example one viewpoint represents all creational activities, responsible for changes made to the artefact. Another viewpoint might represent all activities responsible for the gain of knowledge.

A simple recording of the activities is leading to a linear representation of the creative process, the creative path. However, the creative process is not linear but chaotic. It is necessary to be able to describe the whole complexity of the creative process. As a result, the creativity map was introduced [136]. Through this, it is possible to cover all aspects of the creative process. The creativity map is generated from the creative path. The map creation process transforms the states and transitions of the creative path into their logical relation. Through this it is possible to represent situations where the creator abandoned the current approach and went back to a previous idea.

Creativity maps can be of very large size, depending on the length of the creation process as well as the number of the undertaken activities. The presented approach enables the creation of partial creativity maps. Through this, it is possible to hide parts of the creativity map based on certain attributes of the states as well as a hiding based on the viewpoints or transition types. These attributes can be for example the timestamp or noticeable changes to the attribute itself (e.g. a large increase of words in a document).

States and transitions as well as the, on these applicable, operations share a sound mathematically underpinning and are standardised. This allows the comparison of two or more creativity maps. Moreover, the comparison is not limited to creativity maps belonging to the same domain. The foundation of the creativity maps is always the same, regardless of the domain the artefact is belonging to. This is the first model which enables the analysis and comparison of creative process from different creators and different domains.

Another huge benefit of the creativity map is its adequacy for computer based processing. Other approaches, as discussed in Chapter 2, depend on human processing of the data. This is of course hard and error-prone. Not to mention the enormous amount of time necessary for large projects.

Chapter 4

Pattern-Based Information Extraction

Objectives

- Introduction of behavioural patterns.
- Introduction of the behavioural pattern description language.
- Presentation of the usage of the description language with examples.
- Description of the most common types of behavioural patterns.

“I have had my results for a long time: but I do not yet know how I am to arrive at them.”

Carl F. Gauß (1777-1855)

4.1 Introduction

This chapter will introduce techniques for the automated extraction of information from a creativity map. The thesis is focussing on information, which is represented by the arrangement (constructs) of transitions in the creativity map. These constructs are the mappings of the creative behaviour. The constructs of transitions are from now on referred to as patterns. The patterns are introduced in this chapter as so-called behavioural patterns. A behavioural pattern represents a search pattern, based on the knowledge of the creator and the community. The definitions of those behavioural patterns are user-based and hence provide full flexibility for the analysis of the creative process. A language, enabling machine-readable definitions of behavioural patterns, will be presented. The language utilises an Extended Backus-Naur Form (EBNF) style notation. The behavioural patterns will be applied on the creativity map via a parser. The aim of this chapter is the introduction of an approach, satisfying the need of extracting information about the creative process. Examples are used to demonstrate the flexibility and versatility of this approach. This chapter provides a solution for research question number one: How can information about creative behaviour, which is hidden inside the creativity map, be characterised? As well as partially answering research question number two: Which computer-aided technique enables the extraction of information from creativity maps?

4.2 Data, Information and Knowledge

The process of mapping and analysing the human creativity is divided into three stages. The three stages are best described with the terms *Data*, *Information* and *Knowledge*. The collection of the data is the first stage. It is followed by the creation of the creativity maps. The last stage is the analysis of information, which has been extracted from the creativity map with the help of behavioural patterns. Each stage raises the level of understanding. Therefore, every new stage raises the quality and detailedness of the input for an analysis.

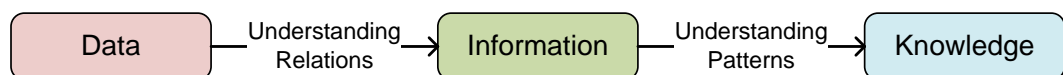


Figure 4.1: Knowledge Extraction

Figure 4.1 displays the positions of the three stages. The data stage represents the raw data which is directly collected from the creator. It only includes states and transitions, which form a linear structure - the creative path. The information stage describes the creation of the creativity maps and partial creativity maps. Unlike the creative path, the creativity map enables the understanding of the logical structure of transitions and states in the data set. The final step is the knowledge extraction. It represents the stage of understanding patterns inside the creativity map, which enable a detailed analysis of the creative behaviour [11].

Every person follows patterns when working creatively. This is mostly a subconscious and hardly noticeable process. The patterns contain valuable knowledge about the creative process of a creator and are crucial for a later analysis. Collecting information about this process enables genuine insights at the creative process.

Obviously, these patterns play an important role for the creative behaviour of a creator. Patterns allow the analyses of the creative process and therefore might be very valuable for the unveiling of the secrets behind creativity itself. Nonetheless, it requires knowledge to be able to interpret the collected information. A framework for the collection, store and sharing of knowledge about creative processes will be introduced in Chapter 5.

4.3 Scope of the Research

This thesis mainly discusses the application of knowledge about creative behaviour for the extraction of information, stored in the creativity map. The focus lies on the definition and application of behavioural patterns, as well as the analysis of the results. The process of transforming data into information is discussed in the thesis of Sascha Westendorf [130]. Hence, for the following chapters of this thesis, it is assumed that the creativity maps and partial creativity maps are already created and ready for the analysis process.

Analysing the creative behaviour of a creator differs from the identification and analysis of information about the artefact. Information like the change of the word count from one state to another is simple gained through accessing the information of those states. The analysis of behaviours, however, requires knowledge about their use. The creator is unable to make assertions about a sequence of states without knowing what useful information can be extracted from it. All knowledge about the creative process is created by people and is based on the experience from previous projects.

4.4 Creative Behaviour

Creativity is often discovered through a simple review of the artefact by experts of the domain. The judgement, whether an object can be considered creative, is simple in regard to the troubles of identifying how the creator mastered to produce the creative artefact. Even amateurs, without a detailed background of the respective domain, can distinguish most objects as either creative or non-creative. For example, the majority of people will agree that the works of Leonardo da Vinci are highly creative. The understanding of what is a creative artefact depends on the society it is received by. However, a society cannot know what has made the artist creative in the first place. The circumstances of what made him creative are usually unknown. This even includes the artist himself. Many artists and geniuses were not able to unveil the secret behind their creativity. The moment of receiving the ideas, and therefore their creative moment is often described as somewhat magical and unpredictable. This phenomenon is not exclusively known for artists, but creators in general.

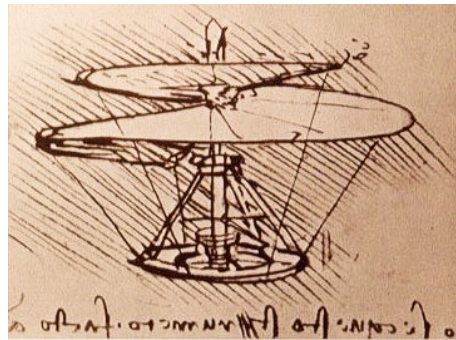


Figure 4.2: Creativity of Leonardo da Vinci (1452-1519)

The key of what makes people creative lies in their abilities and from that derived way of working. This also implies making connections to other people's work. Successful artists, developers, etc. are those, which think and work slightly different than the majority when working creatively. Hence, the most promising way to shed light on the creativity of a person is through the analysis of his creative behaviour. The creative behaviour of the person is the sum of all his activities during the creation of an artefact.

4.5 Behavioural Patterns

The analysis of the creative behaviour is based on knowledge and experience of past projects. This knowledge is not only provided by the creator himself, but usually common knowledge of the community of the same domain. Much of the knowledge is not only useful for a single domain, but interdisciplinary. The knowledge allows making predictions about how a certain creative behaviour will influence the artefact. Based on this knowledge, it is for example possible to search for behaviours which are beneficial for the creative process or others which diminish the creativity of the creator.

It was mentioned before that the creative behaviour is represented by the transitions inside the creativity map and especially by their arrangement. As a result, the analysis of the creative behaviour will be based on patterns, describing combinations of transitions. These patterns are introduced as behavioural patterns in this thesis. The behavioural patterns are used to collect information about the creative behaviour of creators. Therefore, this technique enables the analysis of the creativity maps to search for particular creative behaviour.

Behavioural patterns are applied on the creativity map in order to gain the required information. A parser will traverse the creativity map for that given structure. The result of the parsing process is the information, matching to the description of the behavioural pattern.

The most common behavioural patterns are sequences. Nonetheless, the behavioural patterns also often contain branches.

An example: One could be interested to find out whether a sequence of a *writing*, followed by *discussing*, followed by *writing* appears in the creativity map of the creator. This pattern could for example indicate poor creative behaviour and be a bad sign for the creator if identified in a large number.

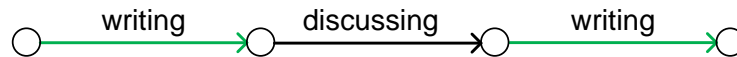


Figure 4.3: Graphical Representation of a Behavioural Pattern

Figure 4.3 depicts the graphical representation of this behavioural pattern. Graphical representations of behavioural patterns are ideal for an easy understanding.

The formal notation of the pattern P is:

$$P = \longrightarrow_{writing} ; \longrightarrow_{discussing} ; \longrightarrow_{writing}$$

Of course, this notation is not very practicable as an input method for a computer-based identification process. Hence, the presented approach uses an EBNF-style language. With this technique, it is possible to define all kind of patterns in a quick and effective fashion.

4.5.1 Length of Sequences

The length of a sequence describes the number of transitions in a sequence, resulting from the application of a behavioural pattern on a creativity map. This information is useful for the explanation of some operators for the pattern description language, which will be discussed later in this chapter.

Given a sequence S , consisting of various transitions:

$$S = \longrightarrow_{writing} ; \longrightarrow_{reading} ; \longrightarrow_{contemplating} ; \longrightarrow_{reading} ; \longrightarrow_{writing}$$

The sequence in the example contains five transitions and therefore a length $l = 5$. This seems to be very trivial, but is very important when comparing two different sequences of a creative path or creativity map.

4.5.2 Length of Results

Every creator changes the working behaviour from time to time. This has obviously an effect on the composition of the creativity map. Changes in the creative behaviour will result into a different order of the transitions. Other changes might effect the length of paths. Hence it is necessary to introduce mechanisms to measure the length of the results from the analysis. The length of the result from a parsing process, containing one or more branches, is defined by the longest possible path. Such a path is a sequence of transitions, which can be followed with a pen without lifting it from the paper. The length of the result is very useful for reducing the result set of an analysis. For example if one is interested in situations where the creator often went back to a previous idea. This situation can be identified in the creativity maps as branches with short sequences.

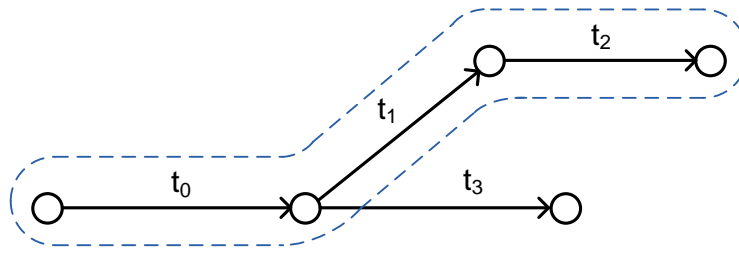


Figure 4.4: Length of the Pattern

Figure 4.4 depicts the result of an analysis. The first transition t_0 is followed by the branch $(t_1 ; t_2) \tilde{+} t_3$. There are two paths or sequences in this subgraph. The first path P_1 consists of $t_0 ; t_1 ; t_2$ and the second path P_2 consists of $t_0 ; t_3$. The longest path in this example is P_1 and has a length of 3. It is marked by the dashed line in the figure.

The later in this chapter presented pattern description language provides an operator for measuring the length. The operator enables to narrow the result set of the pattern to a specified length or ranges (smaller than, greater than and equal to).

4.5.3 Complexity of Patterns

The complexity of a pattern is often related to the length of the results it will produce. A pattern, which will create results with long sequences, is usually more complex than one with short results. The aim of every analysis of creative behaviour is to receive a precise and meaningful result. The process of analysing creative behaviour is based on user-defined patterns. Therefore, it is important to define patterns, which precisely instruct the parser to return the wanted results.

A long pattern with many details might not produce any or just a few results. This can be wanted, but contains the risk of missing valuable information. Long patterns are useful to identify a certain sequence within the creativity map. On the other hand, patterns which produce long results are more vulnerable to the collection of unnecessary information.

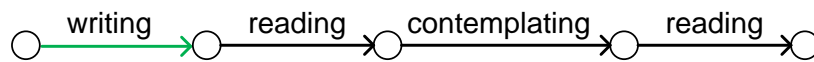


Figure 4.5: A long Sequence

The sequence, depicted in Figure 4.5 is the result of applying a behavioural pattern on the

creativity map. The creator intended to search for all direct successors of the *writing* transitions in a creativity map. However, the pattern expression was incorrect and returned longer sequences. The first two transitions are the answer to the creator's question. The rest of the sequence is useless. Poorly created pattern expressions can lead to mistakes and are often more time-consuming.

4.6 Behavioural Pattern Description Language

The creative mind is not limited in its ways of creating new ideas. Analyses with different foci require different patterns for gaining an useful insight of how creativity was achieved. The identification process for creative behaviour requires flexible and transparent techniques, capable of covering as many situations as possible. A pre-defined, unextendable set of patterns would only restrain the analysis process. Moreover, this limitation would not allow to uncover the full possibilities of the creativity map.

An acceptable solution must provide the following characteristics:

- Simplicity
- Flexibility
- Reliability

All of these three characteristics must be provided in order to be able to mine for the creative process inside the creativity map of a creator. The points simplicity and flexibility are mostly self-explaining. Flexibility was already mentioned and describes the ability to cope with any pattern, based on transitions. Simplicity stands for the fact that the solution must be easy to apply and allow to define new patterns quickly. The point reliability is referring to the extraction process. The technology, which is responsible for the extraction process, must return all through the pattern defined information.

The solution for this problem is a description language for behavioural patterns. The pattern description language, introduced in this chapter, enables an easy and flexible definition of behavioural patterns. It is based on the Extended Backus-Naur Form (EBNF), with some additional operators. More precisely an EBNF with the notation by Niklaus Wirth [133].

4.6.1 Operators and Formal Definition

An EBNF-style notation is very useful for this type of application. It allows describing the behavioural patterns through its basic elements like transitions and operators. The introduction of operators for branching and merging is straight forward. Furthermore, it is possible to define special operators. An example is an operator for the calculating the length of the results from an extraction process.

The following pages of this chapter present the concept of a pattern description language, based on EBNF. The language is called Behavioural Pattern Description Language (BPDF). It is specially designed to express behavioural patterns. This means, it enables the description of sequences and branches, through transitions. It is not supporting states. However, **states** can be taken into account through the use of **partial creativity maps**. The technique is introduced at the end of this chapter.

The following table contains the definition for the elements of the BPDF.

EBNF	Definition
pattern expression	"<pattern>" ":=" ext_sequence ";"
ext_sequence	sequence LEN(sequence)(condition) REVERT() sequence REVERT() REVERT() sequence sequence REVERT() sequence SUM(sequence) LEN(SUM(sequence))(condition) LEN(REVERT())(condition)
sequence	transition {(operator transition) transition} "(" sequence ")" "{" sequence "}"
SUM(sequence)	summarising the results
LEN(sequence)(condition)	length of sequences
transition	" <a_transition>" " <c_transition>" (" <a_transition>" "\ " (" listing ")")
<a_transition>	any-transition

<c.transition>	chosen-transition
REVERT()	revert operator
" ~"	branching operator
" _"	merging operator
" "	or operator
()	grouping operator
{ }	repetition (0...n) operator
\(listing)	without transitions of the listing - operator
listing	identifier { ", " identifier }
identifier	" " " label " " " " <c.transition>"
label	label of a transition
condition	" <" number " >" number " =" number
number	natural number

Table 4.1: Elements of the Language

Table 4.1 provides a brief overview about the operators of the pattern description language. The rules for the parsing process will be presented in detail in the following sections.

Pattern Expression

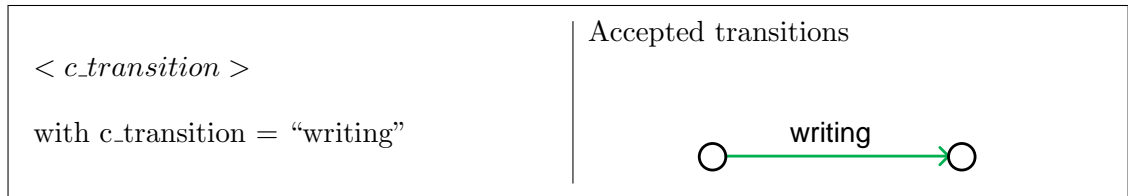
The pattern expression represents the basic structure of a pattern. It is the description of a behavioural pattern with the above explained elements.

$\langle pattern \rangle := ext_sequence;$

The pattern description consists of four basic elements. These elements are " $\langle pattern \rangle$ ", ":", " $ext_sequence$ " and ";". Only the extended sequence ($ext_sequence$) contains the expression of the pattern. The other elements are only indicating the start and end of the pattern expression.

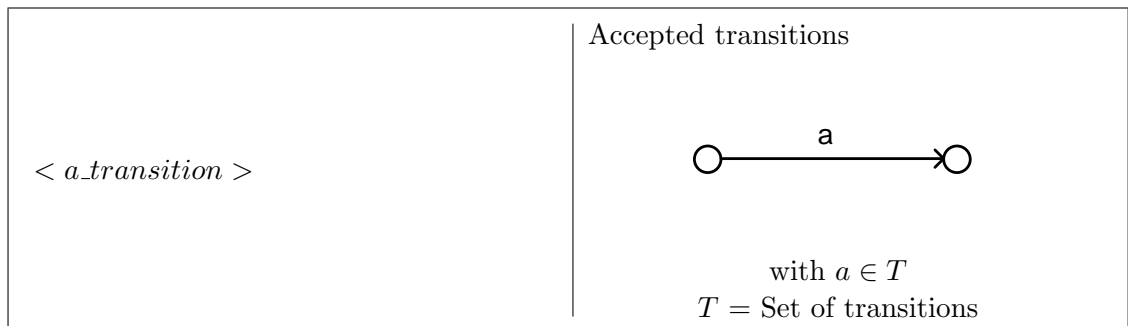
Transitions

Example of a chosen-transition:



The chosen-transition is used to define a transition type at a certain position of the behavioural pattern.

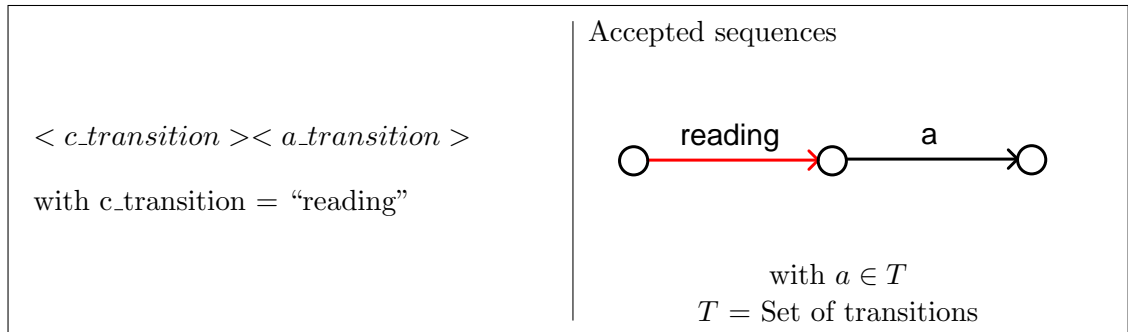
Example of the any-transition:



The any-transition is a placeholder, demanding the existence of a transition of any type at the same position inside the behavioural pattern.

Sequences

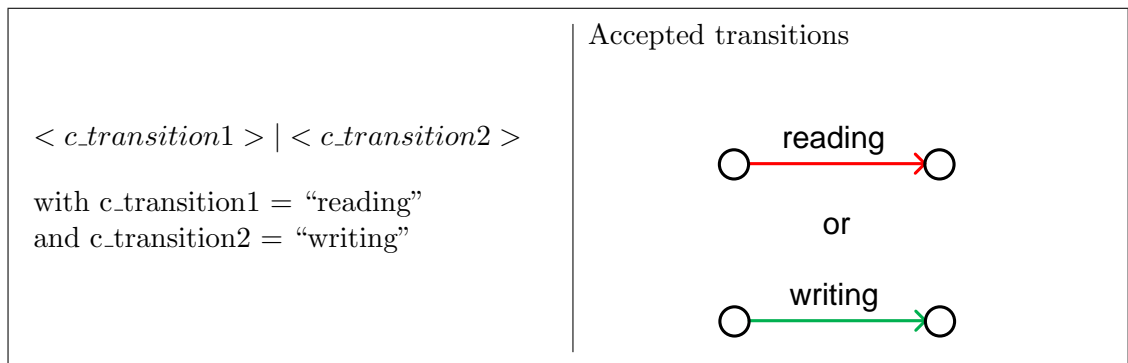
Example of a sequence expression:



Sequences are expressed through a number of transitions, strung together.

Disjunctions

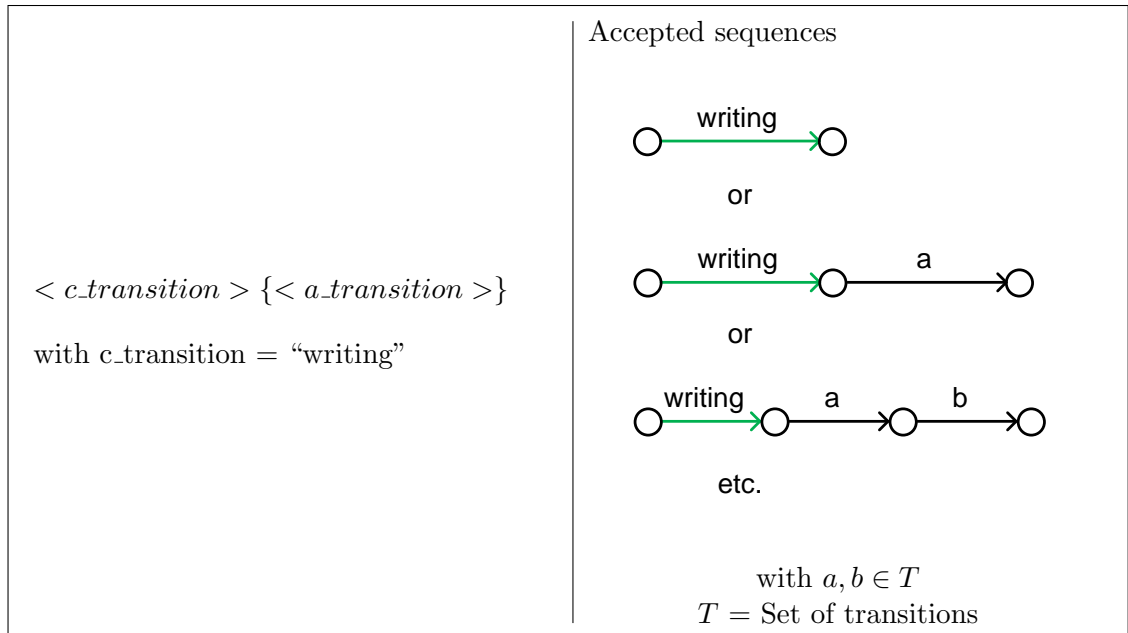
Example of a disjunction:



The disjunction operator of the BPDFL is an *inclusive* disjunction.

Grouping and Repetitions

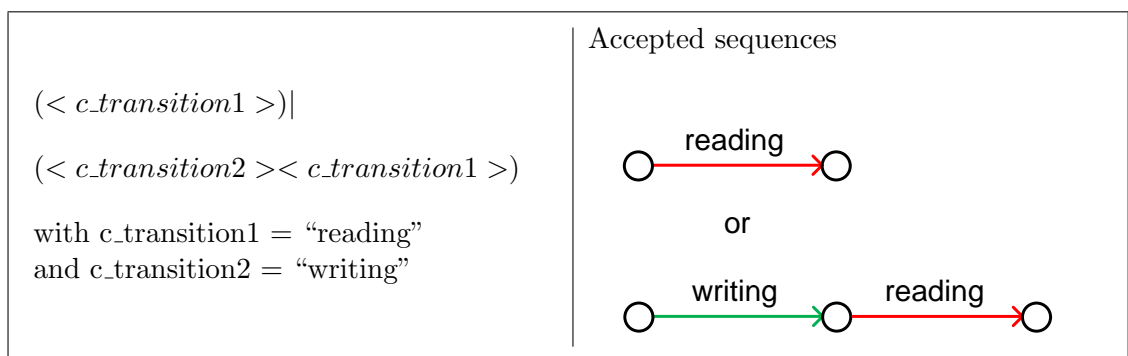
Example of a repetition:



The repetition operator has a range of 0...n transitions.

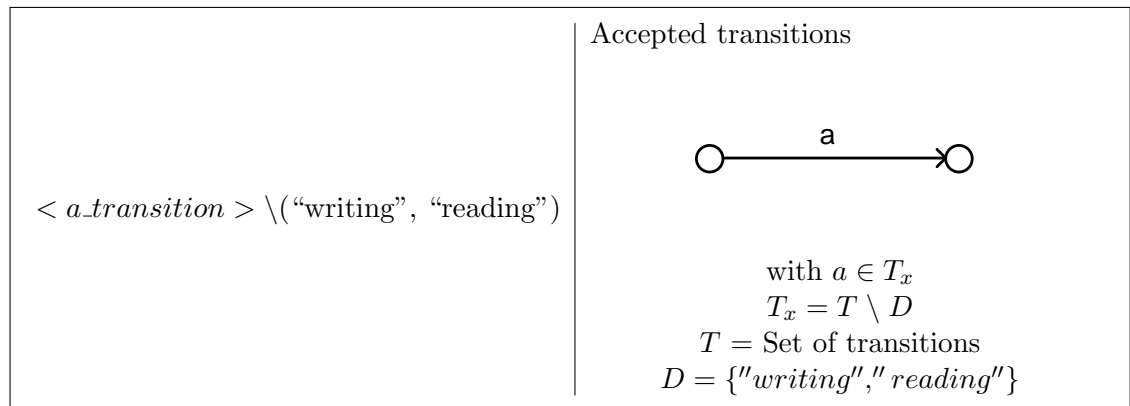
Optionality and Exclusion

Example of optionality:



There is no special operator for optionality like in the original construct of Wirth's EBNF. Optionality is expressed through the | element.

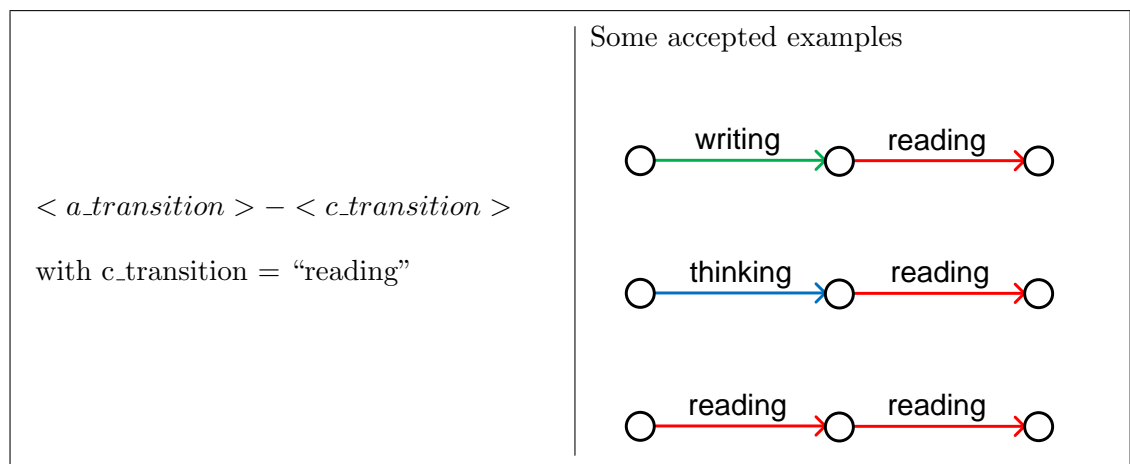
Example of an exclusion:



This operator is useful e.g. for searching sequences without certain transition types.

Choices

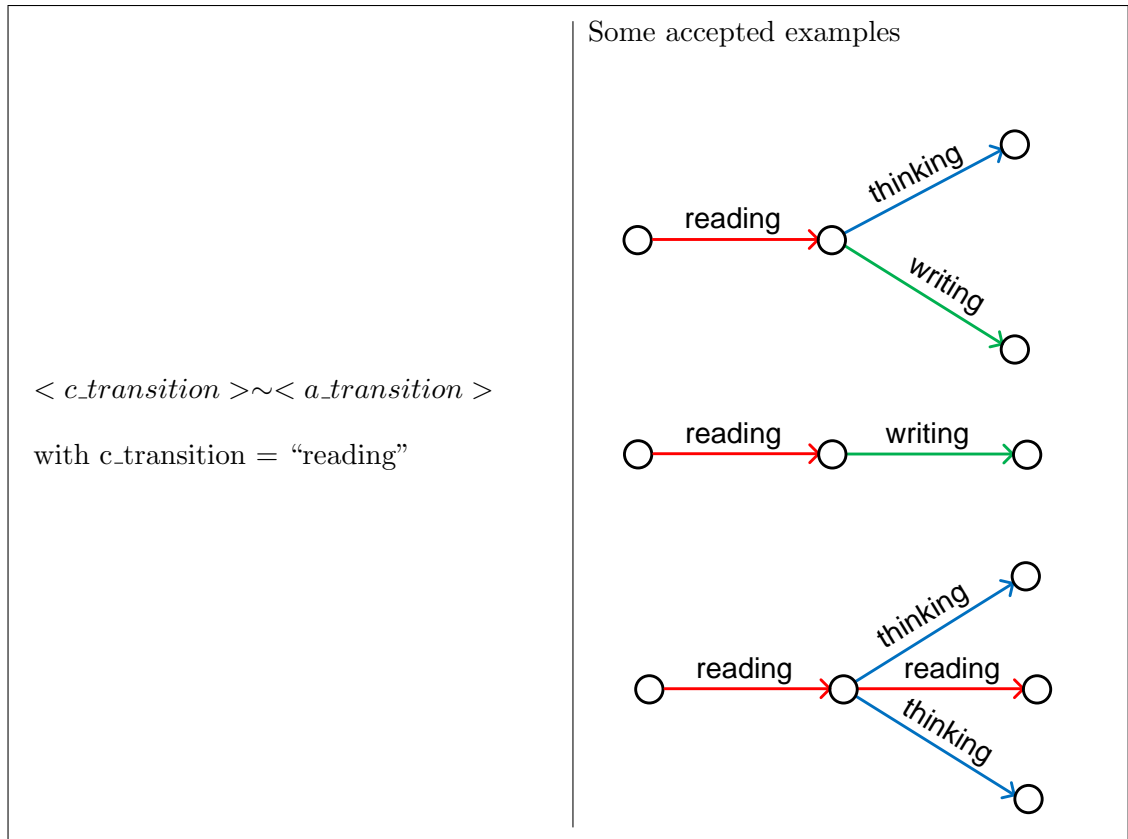
Example of merging:



The merging operator is used for the collection of the incoming transitions for a specified

transition or sequence. NB, this operator is only providing its full usefulness in combination with the SUM operator.

Example of branching:

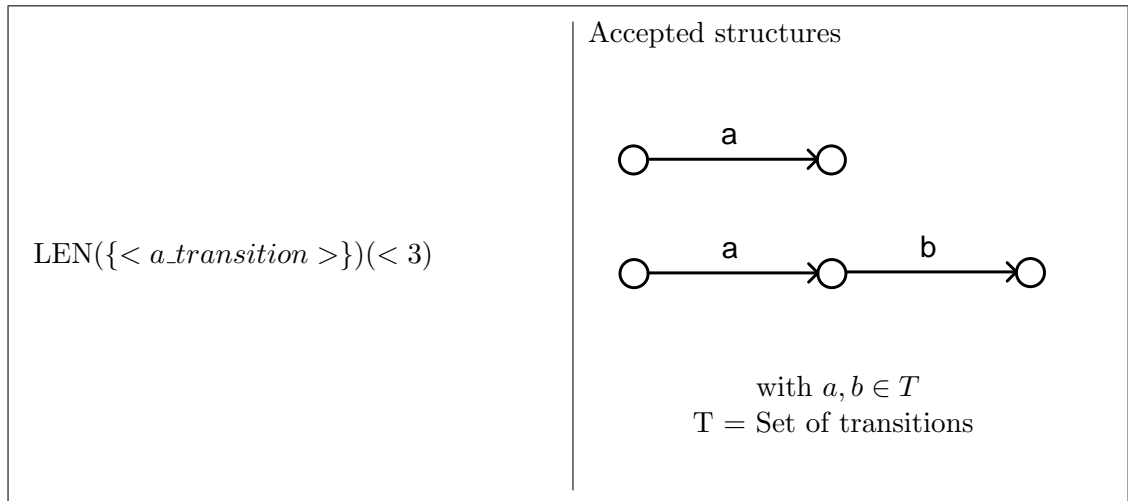


The branching operator (\sim) enables the extraction of branches from the creativity map in order to be able to analyse the choices the creator has made.

Note that both examples for the choices contain only some possible example result entities and not all possible structures.

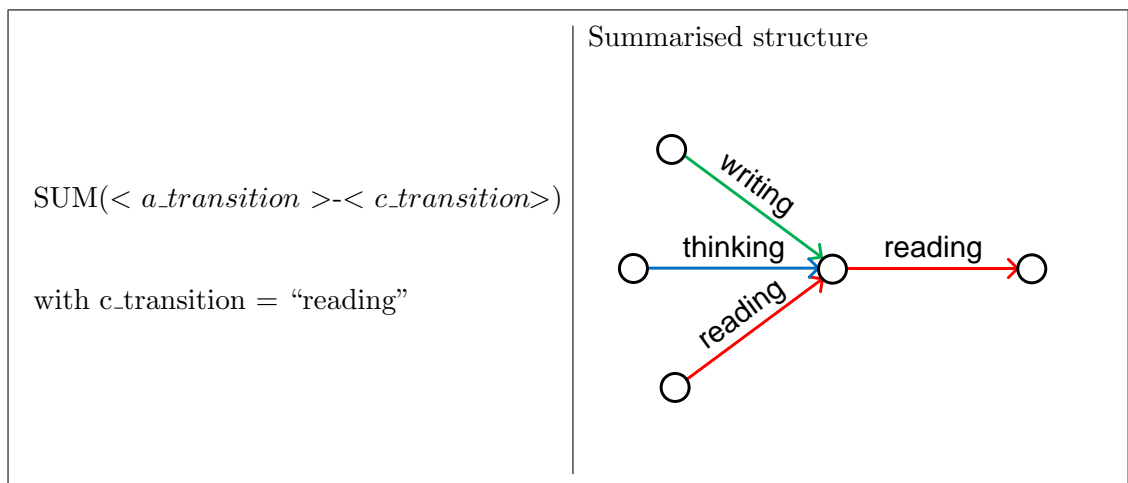
LEN() and **SUM()** Operators

Example of the LEN operator:



This operator is used to create a subset of the result set, containing entities with a certain length.

Example of the SUM operator:



The operator is forming one result entity through summarising all found elements.

4.6.2 The REVERT() Operator

Branches in a creativity map occur where the creator stopped working on the current approach and went back to a previous idea. The previous idea is represented by a state inside the creativity map. The creator will therefore “jump” from the current state to an older state. A new sequence or path will be started from the chosen (old) state after the creator continues working with the resources of that state.

A revert is obviously a very interesting event. This action is worth taken into account for the analysis of the creative process. The event represents the inability of the creator to continue with the current approach. Hence, it might indicate the death of creativity. The analysis, based on this event, provides information about what went wrong and finally left the creator no other choice than re-using an old idea.

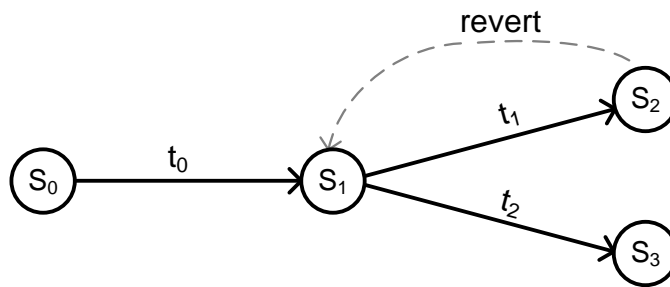


Figure 4.6: Reverting to an old State

Figure 4.6 depicts an example of a revert to a previous state. The creator had ended the path $P = t_0 ; t_1$, with S_2 as the current state, and switched back to S_1 . A new path $P = t_0 ; t_2$ was created after the creator continued working on the in S_1 stored version of the object.

Unfortunately, it is impossible to create a behavioural pattern, which is capable of describing the information about the revert which is needed for the later analysis. The reason for this problem is the non-existence of revert transitions in the creativity maps. These transitions are not stored in the creativity map, because it would make its structure much more complex. Furthermore, the information about the revert operations is already stored inside the states. However, the pattern description language is designed for the search of transitions. It is not capable of handling states. States were purposely not introduced, in order to keep the language and therefore the syntax and semantics simple.

As a result, another solution is required to overcome this problem. Adding an operator is a simple but elegant way to cope with the problem. A new operator, called *REVERT()* was introduced. It is able to retrace the steps which led to a “dead end” in the the creativity map. The information about the sequences is stored during the creation of the creativity map and not afterwards. This is possible, because sequences will not be changed after their creation. New sequences can only be attached at a state as a new a new branch. The information about the sequences can be stored in a database. This increases the execution speed, because it avoids a backtracking of the sequences.

The usage of the operator is quite simple. It can be either used with a $\langle c.transition \rangle$ as parameter or without any parameter.

Pattern: $\langle pattern \rangle := REVERT(\langle c.transition \rangle) \mid REVERT();$

For now, the first parameter can be only a chosen-transition. Future versions will also support sequences or choices of several chosen-transitions. A description of revert patterns at the end of the following section provides more information and examples of the usage. The operator can also handle simple sequences before and after the *REVERT()*, but the elements next to the operator have to be transitions.

The *REVERT()* operator is capable of retracing complex paths. The, for a revert responsible, transitions are often distributed over several branches. Following example will demonstrate the difficulties of the retracing process.

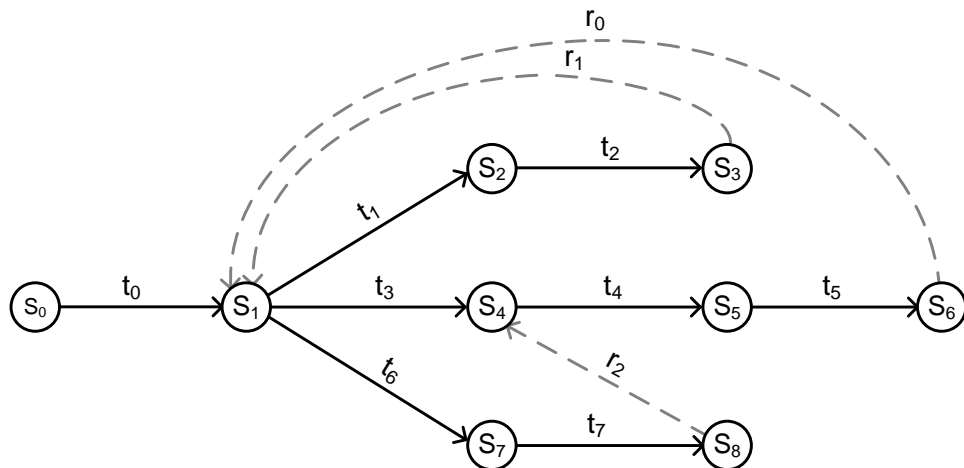


Figure 4.7: Revert Pattern

Figure 4.7 depicts a map with three revert operations. The sequences, responsible for the reverts r_0 and r_1 , are simple. The first revert from state S_3 to state S_1 was triggered because of the sequence $t_1 ; t_2$. The second revert from state S_6 to again S_1 was triggered because of the sequence $t_3 ; t_4 ; t_5$. These sequences are easy to identify. The third revert is more complex. The for this revert responsible sequence is $t_4 ; t_5 ; t_6 ; t_7$. The identification of such sequences is too complex for any type of tree parser.

4.6.3 Parsing Limitations

There is one limitation for this initial version of the pattern language. Parallel transitions (two or more transitions with the same start and end states) are not supported. However, this will not have any effect on the analysis process, because parallel transitions do not exist in the current version of creativity maps.

4.7 Application of Patterns

This section provides some examples for a demonstration of the pattern description language. The examples cover most of the operators and definitely the most complex ones. The examples are categorised by pattern types. The described behavioural patterns are very useful in order to analyse the creative behaviour of the creator. Each type is described with one or two examples. Each example consists of the expression of the pattern in the pattern description language, the results of the parsing process and of course an explanation. The presented approach is of course not limited to the points shown below.

The following examples will be covered in this chapter.

- Sequence Pattern
- Branching Pattern
- Merging Pattern
- Revert Pattern
- Patterns in PCMs

These examples represent the basic pattern types. It is possible to combine them.

The result set of a parsing process will often contain duplicates. This can create confusion for the observer. The parser will therefore show only one copy and the number of the occurrences. This makes the result set more readable and is also beneficial for the creation of statistics.

4.7.1 Sequence Patterns

Sequence patterns have a simple structure. Nevertheless, these patterns enable very interesting analyses. This type of pattern enables the analysis of certain sequences or check for their existence.

One of these sequence patterns is in this thesis referred to as creation pattern. It is clearly one of the most important patterns for the analysis of the creative behaviour of any creator. The pattern describes the actions or activities which lead to a change of the artefact. The aim of the parsing process is the selection of the intellectual activities for a later study of their influence on the creational activities.

This pattern is good for answering the question, why the activities are divided into the creational and intellectual subcategories.

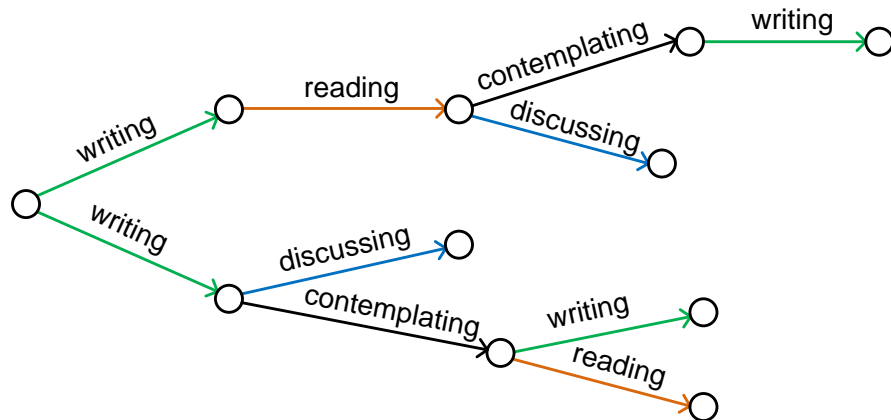


Figure 4.8: A Creativity Map Example

Assuming the creator is a novelist, then the creational activity will be certainly *writing*. Most likely there exists no other type of creational activities inside the creativity map of this creator. Figure 4.8 depicts an example of a creativity map for this scenario. The creational activity are depicted as green arrows.

The description of this pattern through the pattern language is straight forward. The following example depicts the full expression for the sequence pattern.

$\langle pattern \rangle :=$

$\langle a_transition \rangle \setminus (\langle c_transition \rangle) \{ \langle a_transition \rangle \setminus (\langle c_transition \rangle) \}$
 $\langle c_transition \rangle;$

with $\langle c_transition \rangle :=$ “writing”

A sequence of one or more transitions is described through following structure:

$\langle a_transition \rangle \setminus (\langle c_transition \rangle) \{ \langle a_transition \rangle \setminus (\langle c_transition \rangle) \}$

It initiates the parser to find such a sequence. Furthermore, it contains an operator to exclude other occurrences of the chosen-transition. The parser will stop if it reaches one and will return the transitions, which were found so far in form of a sequence. NB, that the chosen-transition can be also used for the without operator.

The extracted sequences, depicted in the following Figure 4.9, are the result from applying the expression from above on the creativity map example, depicted in Figure 4.8.

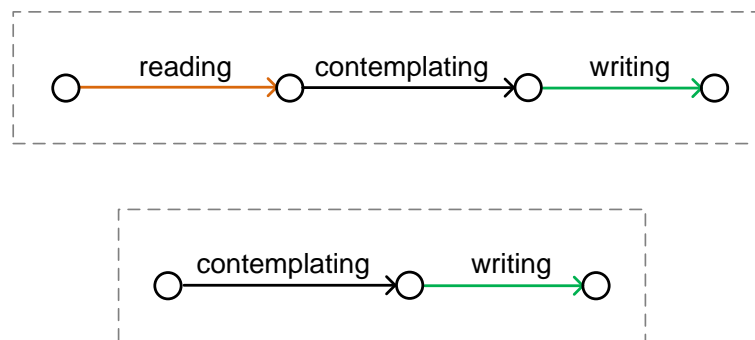


Figure 4.9: The Result of the Creation Pattern

The result set contains two entities. Two of the four writing transitions have no predecessor and are, therefore, not accepted by the pattern expression. Please note that the length of the results is not limited, if there is at least one predecessor.

4.7.2 Branching Patterns

The pattern description language contains a branching operator (\sim) for the description of choices in a creativity map. It is important to be able to cover choices, because creators usually follow a non-linear creative behaviour. The choices occur when the creator went back to an older idea or version of the artefact and started a new creative path. This new path is the new branch, starting from that state.

Following is an example for the usage of the branching operator. The pattern below initiates the parser to collect the direct successors of the chosen-transition. There can be either one direct successor, which would be combined with the chosen-transition into a simple sequence, or multiple direct successors, which create a choice.

$\langle pattern \rangle :=$
 $\langle c_transition \rangle \sim \langle a_transition \rangle;$

The expression for this pattern is very simple. The pattern is defined as a chosen transition, followed by any transition. The length l of the result will always be two: $\Rightarrow l = 2$.

Following a piece of an example creativity map for the application of this pattern. For this example the focus lies on the *writing* activity. Therefore, the writing activity will be the chosen-transition: $\langle c_transition \rangle :=$ “writing”.

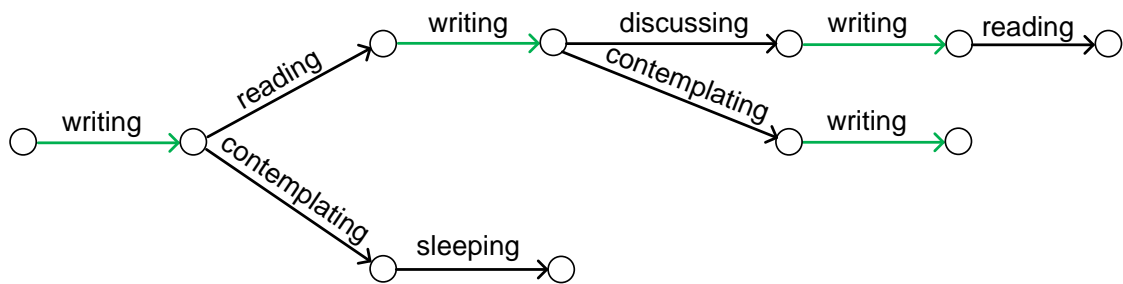


Figure 4.10: Example for the Application of the Branching Pattern

The parser will analyse the example depicted in Figure 4.10. It will search for all transitions with the *writing* label first. Afterwards it will look-up the successors of each of this transitions. The results are depicted in the following figures.

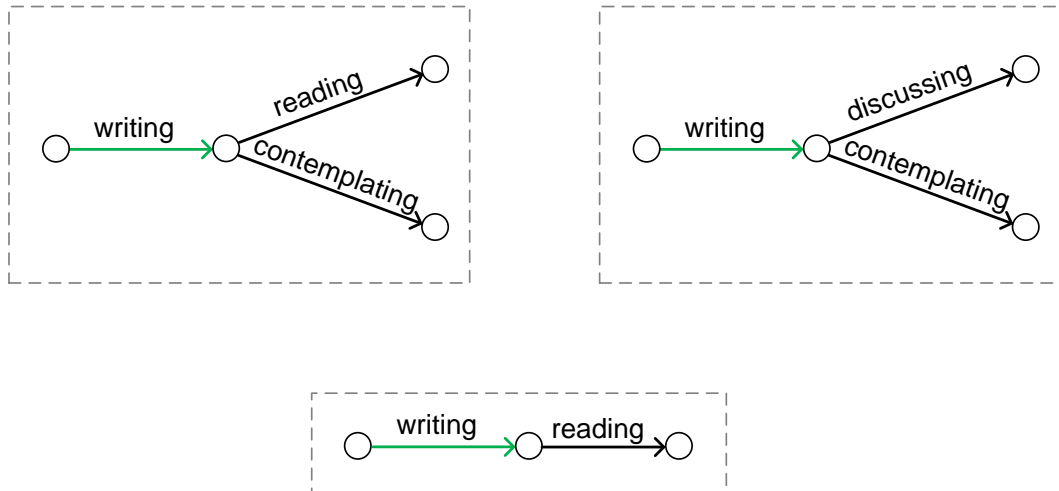


Figure 4.11: Results of the Branching Pattern

The parsing for the choice pattern has produced the results depicted in Figure 4.11. The example contains four *writing* transitions. The parsing process however extracted only three entities. One of the *writing* transitions has no successor and therefore cannot satisfy the conditions set by the pattern definition.

Obviously this pattern extracts the chosen transition with its direct successors. The transitions with only one direct successor are valid for the parsing process as this case is included by the branching operator.

Summarising over Branching Operations

The first example of the choice pattern creates a result entity for each writing element with direct successors. Therefore, the example is not useful for gaining a quick overview. A combination of the branching operator with the *SUM()* operator is very useful to summarise the result for this pattern into one entity, instead of many. All identified successors of the chosen-transition are added to the one entity.

$\langle pattern \rangle :=$

$$SUM(\langle c_transition \rangle \sim \langle a_transition \rangle);$$

The expression for this pattern is only slightly different than the one before. The old expression is used as a parameter for the SUM operator.

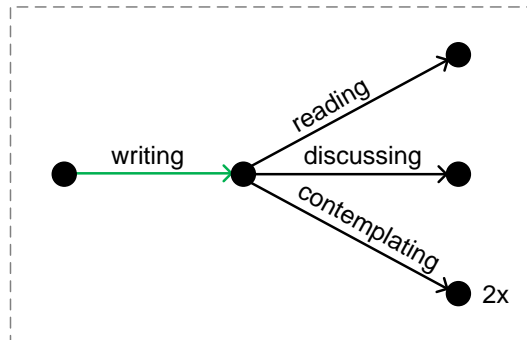


Figure 4.12: Choice Pattern I

Figure 4.12 depicts the result of the parsing process. The pattern was applied on the same example as before. As a result to the SUM operator, the result set contains only one entity for this parsing process. Furthermore, the states in the picture are different. The black states indicate that these are not actual states, but placeholder. This is necessary, as one can see the *contemplating* transition was identified twice. Hence, it can be only a placeholder as the end states of the two transitions cannot be identical.

4.7.3 Merging Patterns

The merging operator is similar to the branching operator. However, the operator is quite limited in its functionality. The reason for this is the fact that transitions cannot have multiple direct predecessors. Nevertheless, the operator enables some very interesting analyses, especially in combination with other operators.

Following example will illustrate the limitation of this operator.

$\langle pattern \rangle :=$

$\langle a_transition \rangle - \langle c_transition \rangle;$

with $\langle c_transition \rangle :=$ “writing”.

The merging operator ($-$) will induce the parser to check the predecessors of a chosen transition. As mentioned before, there can be only one predecessor for each transition. However, the result set usually contains several entities, where the chosen-transition has

different predecessors. The operator is therefore useful when combined with other operators as later explained in the section.

The creativity map, which was presented in Figure 4.10, is used again to apply the above described pattern on it. The chosen-transition $\langle c_transition \rangle$ is the *writing* activity.

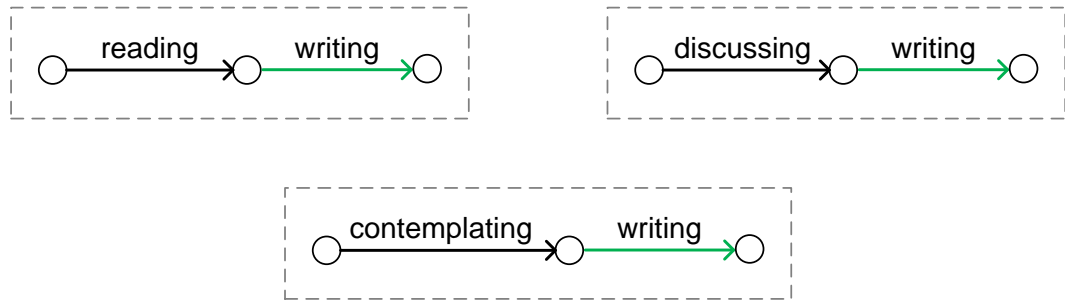


Figure 4.13: Results of the Merging Pattern

The parser extracts only three results, although the creativity map contains four *writing* transitions. The first *writing* transition cannot satisfy the criteria of the pattern and is not added to the result set. The reason for not fulfilling the criteria is the fact that the first *writing* transition has no predecessor. The pattern, however, requires a transition in front of the chosen transition. The first *writing* transition would have been taken into account if the $\langle a_transition \rangle$ had been set optional.

Summarising over Merging Operations

The merging operator is not very useful without the SUM operator. It is possible to receive the same results for the example shown above without the merging operator.

$\langle pattern \rangle :=$

$\langle a_transition \rangle \langle c_transition \rangle;$

The explanation for this phenomenon is the in Chapter 3 mentioned fact that a transition has always just one predecessor. Therefore, the parser is unable to produce results with a choice of two or more transitions. A creativity map never contains a construct which allows a choice of several transitions leading to the same transition.

One might now question whether the merging operator is necessary. The answer to this question is very simple. The operator can produce very meaningful results through the combination with the $SUM()$ operator. Only the combination of this two operators allows to retrieve real merging patterns.

$\langle pattern \rangle :=$

$$SUM(\langle a_transition \rangle - \langle c_transition \rangle);$$

The expression of the pattern is similar to the first example of the merging patterns. The old expression is used as a parameter for the SUM operator, same like in the branching example before.

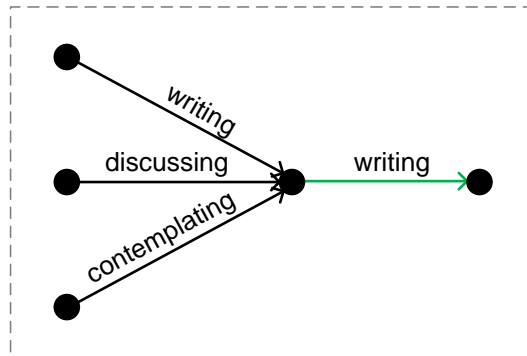


Figure 4.14: Choice Pattern II

The result of the parsing process is depicted in Figure 4.14. The depicted states in the figure are again placeholder and not displaying actual states of the creativity map. In regard to a statistical analysis, the result set is now obviously providing a better overview about the distribution of the transitions, which lead to a *writing* transition.

The most important information provided by the here presented choice patterns is the frequency of every choice. It enables to identify the user's preferences of specific transitions (activities) over others. Moreover, this type of pattern is not limited to sequences with a length of two, but unlimited.

4.7.4 Revert Patterns

The REVERT() operator enables an interesting analysis on what makes creators get stuck with their work. The creativity map contains a lot of information about the creator’s progress making and throwbacks. This information is very beneficial in regard to the development of a creativity support tool. The revert patterns can help to identify those transitions and sequences which have a negative influence on the creative process. The revert patterns are ideal for monitoring the creative process of the creator. In that case, a support tool could warn the creator, if it recognises such unwanted behaviour.

It was mentioned before that the information about the revert sequences is stored in a database and not directly collected from the creativity map. However, this has no effect on the usage of the operator. The user of the framework will not require any additional tools or information. Everything is processed automatically.

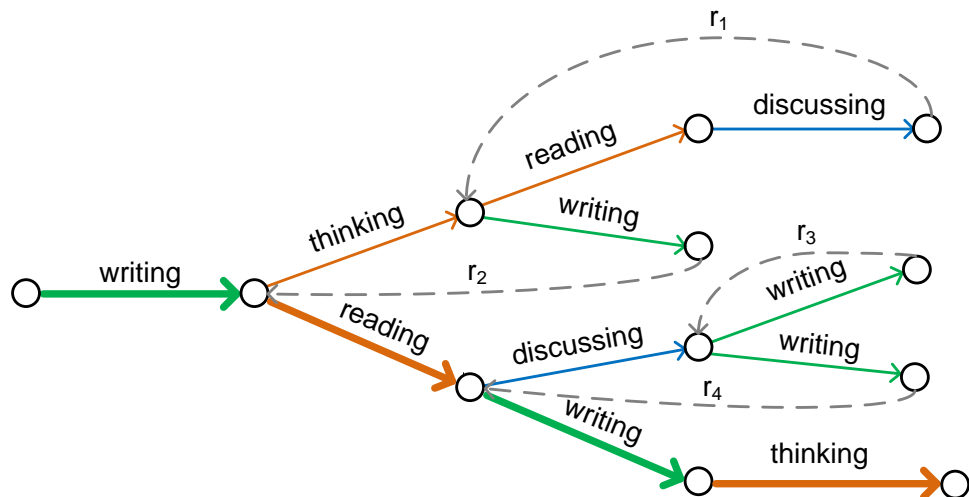


Figure 4.15: Revert Example

Figure 4.15 depicts a part of a creativity map with several branches. The branches are very important for this analysis. As already mentioned, every branch indicates a revert to an previous state. The highlighted path is the current working path. Hence, it is not of interest for the extraction and analysis process. The depicted creativity map contains four revert operations. These operations are depicted in the form of a dashed line. It is necessary to know the target states of the revert operations in order to be able to understand the following examples.

The first pattern will collect all sequences, which have led to a revert operation and contain a *discussing* transition. The expression for the pattern is defined as following:

$\langle pattern \rangle :=$

$REVERT(\langle c_transition \rangle);$

with $\langle c_transition \rangle := discussing.$

The expression for this pattern is very simple. It contains only the $REVERT()$ operator with a chosen-transition as a parameter. The result set is depicted in the following picture.

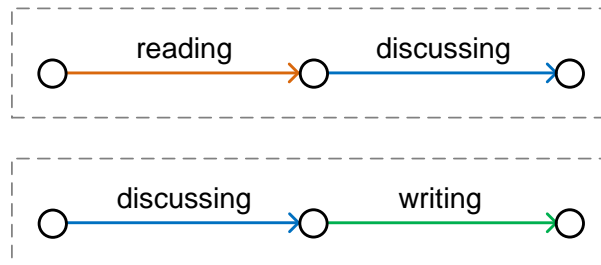


Figure 4.16: Results of the Revert Pattern

The extraction process extracted two entities, based on the two *discussing* transitions of the creativity map. The parameter initiated the parser to search for only sequences with one or more *discussing* transitions.

A second example will demonstrate the combination of the $REVERT()$ operator with the $LEN()$ operator. Such a combination is a good approach for the identification of for example short sequences of unwanted behaviour.

$\langle pattern \rangle :=$

$LEN(REVERT(\langle c_transition \rangle))(= 2);$

with $\langle c_transition \rangle := writing.$

The $LEN()$ operator needs two parameter. The first parameter is the actual pattern description. The second parameter is used to refine the search for revert sequences. In this example the result is limited to the length of two transitions.

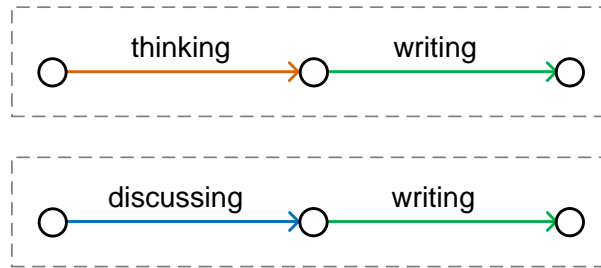


Figure 4.17: Results of the second Revert Pattern

Figure 4.17 depicts the result set of the parsing process. Again, the result set contains two sequences. The second sequence is the same as the second one of the first example. There are only two result entities due to the LEN operator.

4.8 Refinement Through Partial Creativity Maps

The Behavioural Pattern Description Language is designed to enable various types of analysis, based on the transitions in a creativity map. States and their values cannot be described with the description language. The approach has left out states on purpose, as this would raise the complexity of the description language. Otherwise, it could not satisfy one of the main requirements: Simplicity. Another problem arises, if a certain viewpoint should be hidden in order to be able to analysis the correlation of transitions belonging particular viewpoints. The BPDFL itself provides no operator for actively excluding transitions from the information extraction process.

Fortunately, there is an efficient solution for both cases. The in Chapter 3 discussed partial creativity map (PCM) [130] allows the refinement of creativity maps through the hiding of certain information like viewpoints or by state-based criteria. In Chapter 3, the reader has learned that a state is not only representing the artefact, but storing a set of information. The information is stored in a so-called state vector. Hence, the criteria can be one or more entries of the state vector.

Analyses, based on transitions and states, require two processing steps. The first step includes the creation of the PCM with a suitable hiding condition. The second step is the application of one or more patterns on the PCM.

Following is an example of a creativity map. The creativity map belongs to a project of a software developer. The example will demonstrate the creation of a partial creativity map, which hides the transitions of one viewpoint.

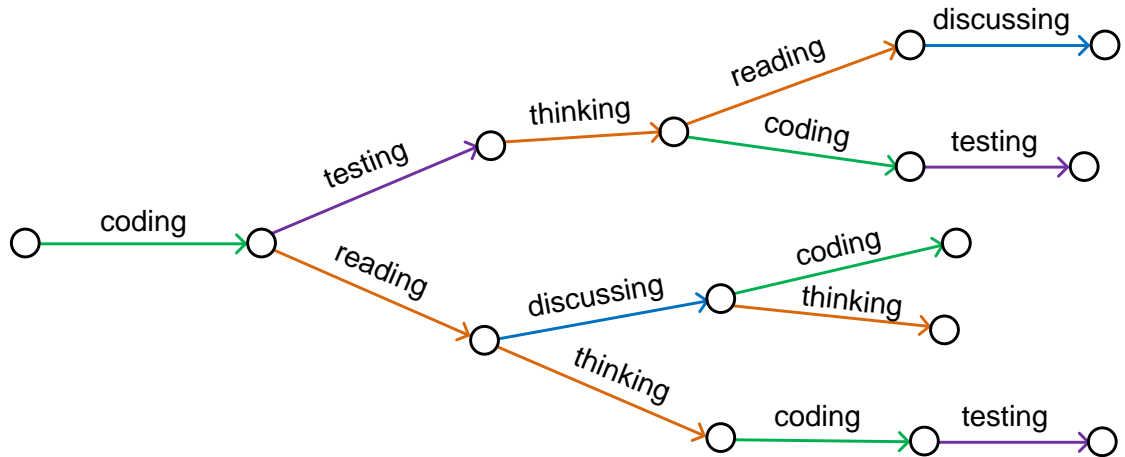


Figure 4.18: A Creativity Map

Four viewpoints can be identified in Figure 4.18. These are:

Creation with $\langle \longrightarrow_{coding} \rangle$,

Knowledge with $\langle \longrightarrow_{thinking}, \longrightarrow_{reading} \rangle$

Test with $\langle \longrightarrow_{testing} \rangle$,

and **Need** with $\langle \longrightarrow_{discussing} \rangle$

The creator might want to analyse the creativity map without the viewpoint *Creation*.

This will result in a new Partial Creativity Map P :

$$\langle P, \longrightarrow_P \rangle = \langle T, \longrightarrow_T \rangle \times \left(\bigcup_i \langle A_i, \longrightarrow_{A_i} \rangle \setminus \bigcup_j \langle H_j, \longrightarrow_{H_j} \rangle \right)$$

where T is the temporal viewpoint, A the set of all viewpoints and H the set of hidden viewpoints ($H = \langle Creation, \longrightarrow_{Creation} \rangle$):

The PCM is depicted below. All four *coding* transitions have been removed.

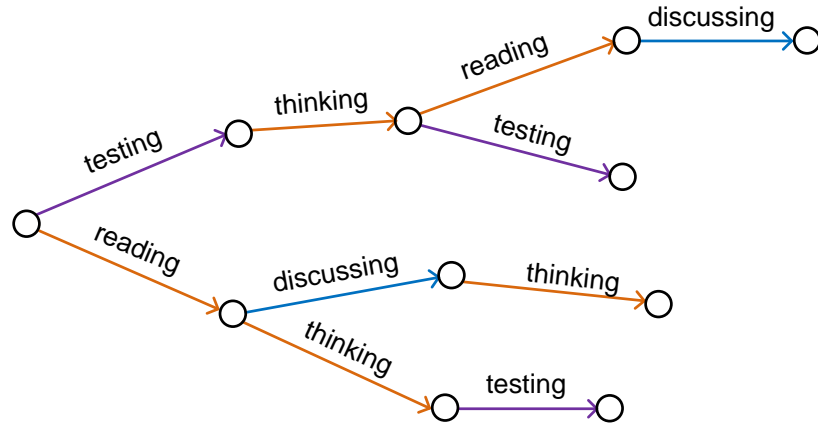


Figure 4.19: A Partial Creativity Map

The creator can now extract the information he requires. The following expression is used to extract a sequence from the PCM.

$\langle pattern \rangle :=$

$\langle c_transition1 \rangle \langle c_transition2 \rangle ;$

with $\langle c_transition1 \rangle := "thinking"$ and $\langle c_transition2 \rangle := "testing"$.

The pattern is defined as a *thinking* transition, followed by a *testing* transition.

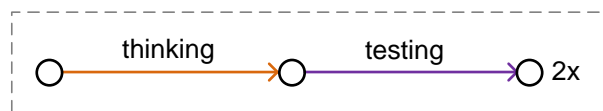


Figure 4.20: Results of the Extraction Process

The result of the extraction process is depicted in Figure 4.20. The sequence has been found twice inside the PCM. It is important to note that one of the sequences would not have been found if the expression had been applied on the normal creativity map. The reason for this is that the *coding* transition on the bottom of the figure has been hidden.

An analysis of a PCM, based on an entry of the states, would be identical to the example above. The presented approach enables a huge variety of analyses. It includes states into the analysis process as well as the interaction of states, belonging to one or more viewpoints.

4.9 Summary

The first pages of this chapter provide information about the scope of the thesis. It is explained that the presented research is part of the Creative Technologies Research Programme at the Software Technology Research Laboratory (STRL). The creation of the creativity maps is not part of this thesis, but of another. Hence, it is always assumed that the techniques for the creation of those are already provided.

The concept of behavioural patterns is introduced in this chapter, in order to be able to characterise creative behaviour within the creativity map. The number and structure of behavioural patterns is not limited. The approach represents an open technique which can be adapted to the needs of the person who is analysing the creativity map. A behavioural pattern is therefore not only used to describe a creative behaviour, but also supports the search for creative behaviour.

This approach is the solution for research question number 1: How can information about creative behaviour, which is hidden inside the creativity map, be characterised?

Even though, it is possible to analyse the creativity maps by hand. The real strength of the approach emerges with a computer-supported analysis process. A precondition for a computer support is the ability to describe the behavioural pattern in a structured and computer-readable way. This requires a flexible and reliable solution. A description language is introduced in this chapter as an elegant and efficient solution.

The so-called Behavioural Pattern Description Language (BPDFL) provides all necessary facilities for a successful application of behavioural patterns. The chapter provides detailed information about the structure of the description language. The description language uses a EBNF style notation. Descriptions of behavioural patterns can be defined quickly and easily. This satisfies the requirement for a simple, flexible and reliable solution. The language is not limiting the number of patterns. New patterns can be described easily. As a result, even people with no knowledge of programming languages are able to define behavioural patterns with the BPDFL.

Some of the most important types of behavioural patterns have been explained with examples. The parsing and extraction process works with normal creativity maps as well as with partial creativity map. The latter type makes it possible to refine the input for the parsing process. This allows an analysis, based on states, even this is not directly provided through the behavioural pattern description language.

The architecture of a parser for the Behavioural Pattern Description Language is described in Chapter 7. It represents the second part for answering research question number two: Which computer-aided technique enables the extraction of information from creativity maps?

Chapter 5

Knowledge Management and Store

Objectives

- Introduction of a knowledge repository for the management and store of the behavioural patterns.
- Description of the content of the knowledge database.
- Description of the functionality of a knowledge management facility.
- Introduction of techniques for coping with synonymy of terms.

“It is beyond a doubt that all our knowledge begins with experience.”

Immanuel Kant (1724-1804)

5.1 Introduction

Behavioural patterns are based on the experience and knowledge about creative processes. New knowledge is derived from the already known in a constant process. This knowledge is certainly not only the knowledge of a single person, but of many people from the same or other domains. It is necessary to store the knowledge about creative processes and creativity in a central database. The database must enable the management of all the behavioural patterns.

This chapter is presenting a concept for managing knowledge, consisting of behavioural patterns and their associated information. The facilities of the framework enable the access, adding and editing of the knowledge entries. It is also possible to autonomously search and compare behavioural patterns of different users. Through this, it is possible to identify common behavioural patterns. The content of this chapter represents the solution for research question number three.

5.2 Knowledge Repository

The behavioural pattern description language is only useful, if the creator has access to a set of the, for him important, behavioural patterns. It is necessary to provide a facility to store and organise the through the patterns represented knowledge. Furthermore, all behavioural patterns are based on the experience of the creator. Therefore, behavioural patterns can change, or new can be created. A storage facility with a fixed set of behavioural patterns cannot cope with every aspect of the creative process. There is also a requirement for the storage facility to guarantee workplace independency. People, who work with computers, often use two or more of them, when working on a project. Hence, there is a demand for a technology which can provide the same data on all used computers. This can be only provided through a centralised storage facility. The behavioural patterns itself need additional information in form of descriptions in order to be able to apply them to the creativity map and to allow the correct interpretation of the results. There must be a description for each behavioural pattern. The knowledge repository must handle this information as well.

This chapter introduces a storage framework for the behavioural patterns and their associated information. The storage framework is a so-called knowledge repository. The knowledge repository is the central database for behavioural patterns. The database is not a simple storage for the pattern expressions, but contains a lot of information about their meaning and for which type of analysis they can be used for.

The knowledge about the behavioural patterns is not only limited for the use of a single creator. Some of the patterns might be universal for all people of the same domain or even for people from other domains. A framework for the management of the behavioural patterns must support the exchange of patterns. There is a huge demand on collecting and distributing knowledge for a detailed and successful analysis of the creative process [118], especially in a collaborative environment.

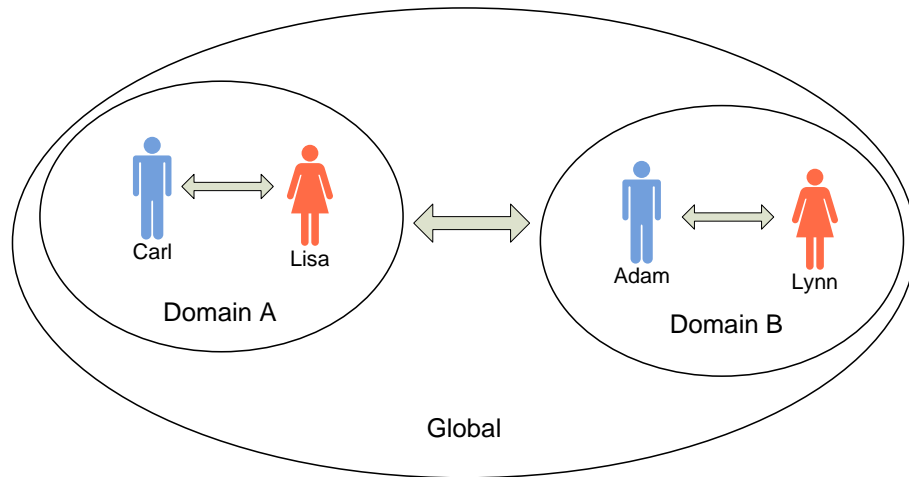


Figure 5.1: Exchange of Knowledge

Figure 5.1 depicts the levels for the communication of knowledge. The arrows in the figure indicate the exchange of knowledge. There are two levels on which creators will exchange their knowledge. The closest circle of exchange is between people of the same domain. The second and loosest circle is the exchange of knowledge between domains. The reason that there are no arrows between people from different domains is because of the management of the knowledge. Firstly, a direct exchange of patterns between two creators would quickly become chaotic and confusing with a growing number of users. Such an approach would be rather destructive than helpful. Secondly, behavioural patterns are categorised, depending on whether a pattern is used in all domains, in one domain or only used by one person. People can only directly add behavioural patterns to their personal set. Those patterns can only move up to a higher level, if they are found in a certain number inside the current level. An example would be a behavioural pattern, which was identified in 30 percent of all personal sets. The owners of the pattern could then vote, if the pattern should be added to the domain level.

5.2.1 Knowledge Categories for Behavioural Patterns

It was already mentioned that it is possible to categorise the behavioural patterns due to their influence on other creators and domains. Some behavioural patterns are found in every domain and some in opposite are very specific for one user, but not useful for any other. There are three categories: Common Behavioural Patterns, Domain-Specific Behavioural Patterns and Personal Behavioural Patterns. An own category for behavioural patterns of a project was not introduced. Patterns of this category would also either fit to the personal or the domain-specific category. The categories would not be isolated from each other anymore and lead to an imprecise classification.

Common Behavioural Patterns

Common patterns are those, which are applicable for every domain. These represent the common knowledge about the creative process. The patterns of this category are useful for the majority of creators from all domains. In some cases it could provoke negative effects for the creativity of a user. Hence, every behavioural pattern needs to be studied before applying it on the own creativity map.

Domain-Specific Behavioural Patterns

Domain-specific patterns describe the knowledge of one domain. The pattern will be found in most of the creativity maps from creators of that domain. This type of behavioural patterns is much more focussed on the background of the creator. For example, software developers share a lot of behaviours when creating a new application. However, the problem of possible negative effects through the usage of unsuitable patterns can occur as well, even if in presumably much smaller number of occurrences.

Personal Behavioural Patterns

Personal patterns are very specific behaviours of a single person. These behaviours will not appear in most of the others creative behaviour. This type of patterns was only created for the creators's needs. One could say that these patterns are "tailor-made". On the other hand, this does not imply that a personal behavioural pattern cannot be useful for other creators.

Knowledge Hierarchies

The behavioural patterns in those three categories are organised in a hierarchical structure. Patterns from different categories will not have the same priority for the creator. For example, a creator identifies a certain pattern in his creativity map. The pattern is described in the domain-specific pattern database as well as in the common pattern database. It is now possible that the appearance of this certain behavioural pattern is the sign of poor creativity in the list of the common patterns and a wanted behaviour in the list of the domain-specific patterns. The question is, which of them has a higher priority for the creator?

The answer to this question becomes clear when the three categories are compared. The knowledge of the common category matches the behaviour of many people. The knowledge of the domain-specific category is more focussed on the conditions of a domain. Nonetheless, the top priority knowledge is represented by the personal behavioural patterns. These patterns overrule all other patterns, because they specifically take the habits (either good or bad) of the user (creator) into account.

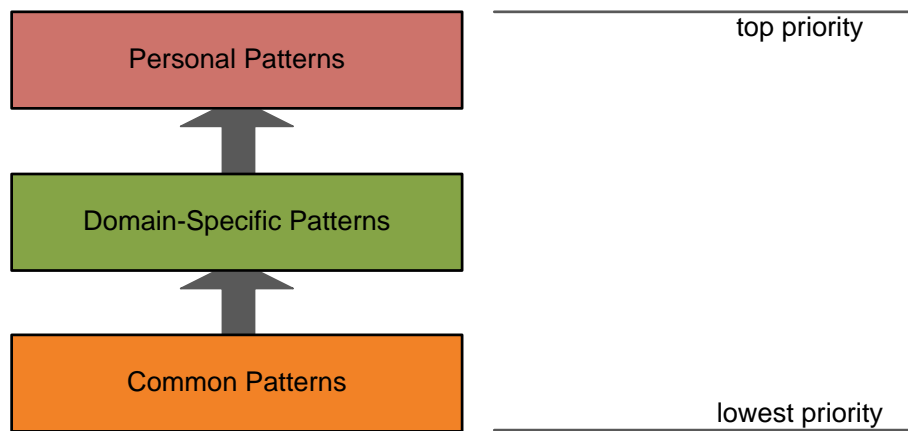


Figure 5.2: Hierarchy of Patterns

The figure above depicts the hierarchy of the behavioural patterns. The lowest priority category is on the bottom of the figure. The arrows indicate the raise of the priority with each higher level.

A situation could occur, where the personal knowledge repository contains the wrong information about a behavioural pattern and the common knowledge repository the correct.

One might argue, that the common knowledge should be preferred in this case. However, this is not the case. The creator of the personal behavioural pattern should be only informed that there is a similar pattern with a different information. He must then check, whether the description of his pattern is correct or not. The creator himself must cater for the correctness of the information, which is describing the behavioural pattern.

5.2.2 Metainformation

A database, containing only the expressions of the behavioural pattern would not be very useful. It was already mentioned, that the database needs also to store information about the behavioural patterns. This type of information is so-called metainformation, also know as metadata.

The US National Information Standards Organization describes metainformation or metadata as following in their manual Understanding Metadata [99].

“Metadata is structured information that describes, explains, locates, or otherwise makes it easier to retrieve, use, or manage an information resource. Metadata is often called data about data or information about information.”

The behavioural patterns represent the information about certain behaviours and the metainformation provide information about the application of those behavioural patterns. Furthermore, a description about the purpose of the results, which is returned through the extraction process.

$$\textit{Knowledge} = \underbrace{\textit{Behavioural Patterns}}_{\textit{Information}} + \underbrace{\textit{Metainformation}}_{\textit{Purpose}}$$

As a result, the knowledge is created through the combination of the behavioural patterns and the metainformation. It is, therefore, the combination of the extracted information with the purpose of extracting that information.

Another benefit of using metainformation lies in the possibility to search for keywords. For example, the set of patterns for a domain can grow very large. It would take a long time to review the all the behavioural patterns in order to find a few suitable ones. Metainformation enable a powerful search for specific behavioural patterns.

5.2.3 Structure of the Knowledge Repository

One can see that this project has very specific requirements for a knowledge repository. Hence, it is necessary to develop a new system, instead of just using already existing approaches. Current knowledge repositories are often developed for the collection of corporate knowledge. These systems cater for the management and categorisation of knowledge inside documents. The aim of these systems is a database, containing all documents of a company, where metainformation is represented through standardised tags. An example for the specification of metainformation is the Dublin Core. It describes the conventions for handling metainformation, especially for documents [86]. In contrast, the knowledge about the creative processes has a narrow input variety. Nonetheless, the associated metainformation is much more detailed and also part of the knowledge itself, not only describing it.

The knowledge repository for managing the knowledge about the creative processes consists of two major facilities. The first one is the knowledge management facility (KMF). It enables the users to access and manage the knowledge. The second facility is the knowledge database. The database stores all behavioural patterns and their metainformation.

The two facilities cater for several different tasks. These are:

- Knowledge Management Facility
 - Knowledge Creation
 - Knowledge Transfer
 - Knowledge Utilisation
- Knowledge Database
 - Knowledge Retention

The tasks are mostly self-explaining. The first task, Knowledge Creation, describes the ability of adding new knowledge to the knowledge repository. Furthermore, it describes the changes of the behavioural patterns and metainformation. Knowledge Transfer describes the communication of knowledge between creators. It includes the conversion, filtering and communication of knowledge. The point Knowledge Utilisation describes the access to the knowledge as well as the search for knowledge. These three tasks are all related to the KMF. The fourth task belongs to the knowledge database. It is called Knowledge Retention. This facilitates the persistent storage of the behavioural patterns and the metainformation [97].

The combination of the knowledge database with the KMF enables the minimisation of redundancy issues. Furthermore, it will cater for the freshness, completeness, consistency, trustfulness and precision of the knowledge. The points are achieved through the multi-user approach. The common and domain-specific behavioural pattern will be available to a large group of creators. These creators will discuss the patterns and therefore aim to identify possible mistakes and to complete the metainformation.

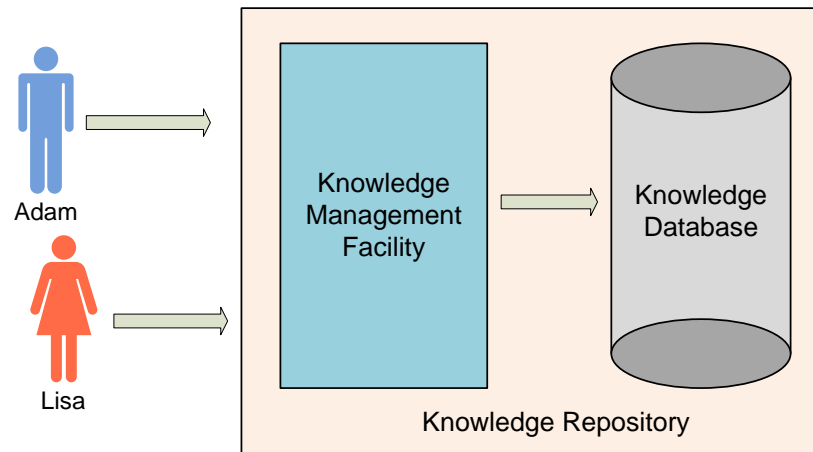


Figure 5.3: The Knowledge Repository

The architecture of the knowledge repository is depicted in Figure 5.3. It is based on a two-tier architecture, consisting of a logic and data tier. The logic tier is represented by the KMF. It is responsible for the structuring and management of the knowledge. The data tier is represented by the knowledge database. It represents the store for all knowledge. The knowledge repository is not responsible for the presentation of the knowledge. This enables a modular architecture and makes it possible to use different graphical user interfaces for the presentation and use of the knowledge. Hence, the two people in the figure represent creators, using a front-end software to access the knowledge repository. Creators will never directly access the knowledge database. The communication of patterns is only possible via the KMF. This facility manages the access to the database, in particular user access rights and the processing of new or existing behavioural patterns and metainformation.

The basic structure of the presented knowledge repository is not different from other approaches. Differences exist mainly in the KMF. The facility is not only responsible for managing knowledge or adding new knowledge. It caters actively for the exchange of knowledge between the creators.

5.2.4 Knowledge Database

The knowledge database is the central storage of the knowledge repository. A central database has several advantages in opposite to local storage facilities, like text files or even a local database instance. The database is located on a separate server. The access to the data itself is therefore workplace-independent. Furthermore, a normal user is not able to delete the whole database, neither by accident nor deliberately.

Another disadvantage of personal or distributed knowledge databases is the requirement of a much more complex knowledge sharing and management facility.

“ It is virtually impossible to maintain synchronicity among the knowledge bases that require the same knowledge. Without a single source repository, the costs of other maintenance tasks (e.g., knowledge verification and validation) are not easily shared across knowledge bases.” [132]

This addresses the management of the domain-specific and common knowledge. A central knowledge repository enables an easy and flexible way of collecting and distributing behavioural patterns and the related metainformation. Local repositories, like files, would require much more effort regarding the distribution of knowledge.

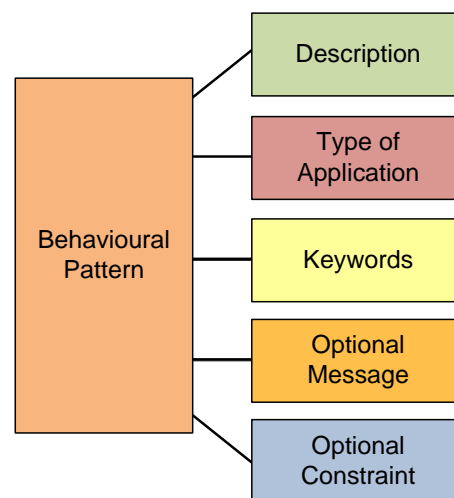


Figure 5.4: Elements of a Knowledge Entry

The Figure 5.4 shows the elements of a knowledge entry. The behavioural pattern is connected with a number of metainformation. These are the description of the behavioural

pattern, the type of application, keywords, an optional message and an optional constraint. The meaning of the description has already been explained.

The type of application defines, in which analysis mode it should be used. An analysis can be performed at two different stages. One stage is the constant monitoring of the creative process. In this case, the user is constantly monitored during his work. The second stage is a detailed analysis of the creativity map, which is only performed sometimes. Some behavioural patterns might be only useful for a monitoring process, others for a detailed analysis or suitable for both cases.

A set of keywords is used for a classification of behavioural patterns. It simplifies the search of patterns for a certain task. Moreover, it eases browsing through particular categories. The last two entries of the metainformation are optional. An optional message is used in case of a monitoring of the creative process. The message can be used to inform the creator, in case of e.g. poor creative behaviour. For example, if a behavioural pattern, describing poor creative behaviour, is recognised. The other entry is the optional constraint. A constraint is a description of the parameter for the creation of a partial creativity map (PCM). With other words a constraint describes the criteria by which the PCM is created. For example, a constraint could describe the hiding of one or more transition types.

5.2.5 Knowledge Management Facility

The knowledge management facility (KMF) represents the logic tier of the knowledge repository. It enables the creator to access the knowledge database. Furthermore, it is the only facility which directly accesses the knowledge database, due to management and security reasons. An included access management caters for the security of the database entries. Its interfaces enable other tools, like the analysis tool, to request the pattern descriptions and the related metainformation. The facility provides further functionalities. It enables the search for behavioural patterns and keywords. It is actively supporting comparisons of creativity maps through automatical categorisation of transition types.

Knowledge is never static, but subject to constant growing and changing. The growing of the knowledge is not only related to a rising number of system users. The normal users will create more and more knowledge, mainly when working on new projects. Updates are very important regarding the freshness of the knowledge. The activities around the management of the knowledge can be best described as a cycle. This is similar to the lifecycle of knowledge as described in Chapter 2. Nonetheless, the circle provides a better illustration of the process in regard to the creation of new knowledge through the learning process during the analysis of the creative process.

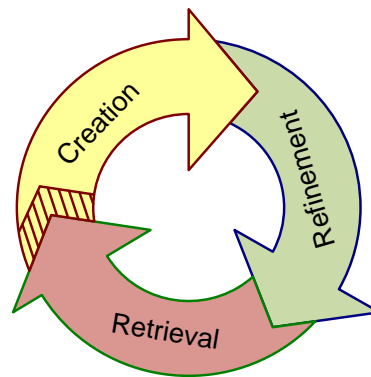


Figure 5.5: Cycle of Knowledge Management

The Figure 5.5 represents the cycle of knowledge management. It starts with the *Creation* arrow. The cycle can end with the *Retrieval* arrow, but it is also possible that new knowledge is based on existing knowledge. This describes the possibility of creating an entirely new behavioural pattern or a new one, based on existing patterns. The *Refinement* arrow represents possible knowledge refinement within the KMF.

The KMF guarantees that newly added information fits into the structure of the knowledge repository and in particular the knowledge database. This prevents the need of complex processes to normalise the knowledge such as in other knowledge repositories. A standardised structure is very important for the analysis tools. These tools require one format which allows them to represent the knowledge about the behavioural patterns and the results of the parsing process to the creator. Some knowledge also enables the analysis tool to trigger an action when a certain pattern is identified.

The KMF also provides functions for the exchange of behavioural patterns. This makes it also to a communication device. The facility improves the communication of behavioural pattern between the users of the knowledge repository. It was already mentioned that behavioural patterns will only move up to the common and domain-specific categories, if the same pattern has been identified in several of the personal pattern lists. The final decision about the upgrade of a personal behavioural pattern to one of the two categories is made through a vote. Those votes are also initiated by the KMF.

Knowledge Retrieval

The main task of the KMF is the retrieval of information from the database. It will select the required entries of the database and return the knowledge in a, for the user appropriate, structure. The complexity of the database enquiries will be hidden. The ability to search for keywords was already mentioned. This is an important feature for the users. It enables an overview about behavioural patterns of certain field of application.

The knowledge access and search facilities are usually accessed through analysis tools. These tools can use the KMF through defined interfaces. The KMF provides access to the management facilities. Users will not be able to access the personal knowledge of others.

Knowledge Creation

A knowledge repository supports a constant growing of knowledge. The system must be able to cope with the changing requirements over time. The approaches for the analysis of the creative process will change or be replaced by new ones. Hence, it is necessary to provide clean interfaces, allowing the users of the system to add new content.

The in this chapter presented knowledge repository is designed to facilitate and encourage the growth of the stored knowledge. It structures the knowledge and makes sure that all required information is present, before writing the knowledge set into the database.

Knowledge Refinement

The knowledge entries of the database can contain errors or require a refinement. Therefore, it is not only important to be able to add new knowledge, but to change already existing knowledge. The KMF enables the manipulation of the knowledge inside the knowledge database. The knowledge entries can be accessed through a well defined interface, like for the knowledge retrieval. This allows a comfortable process for changing a knowledge entry.

However, there are some limitations. It is impossible for one user alone to change or even remove a knowledge entry of the common or domain-specific category. Otherwise, one person could temper with the analysis process of all creators, using this knowledge entry. As a result, only personal knowledge can be changed.

Changing the knowledge entries of the two other categories is not a useful feature. This would be even the case, if such an operation would require the approval of a group. Some of the creators might want to keep using the current version of the knowledge entry. This would end in a conflict of interests. Even a large majority should not be able to change the entry for those people. Otherwise, it is possible that some people might not realise that the knowledge entry has been changed. This would lead to problems and false results of the analysis.

5.2.6 Access Management

The knowledge repository contains an access management system to organise and secure the knowledge entries. Each user needs to create an account before start using the system. This common approach enables a wide spectrum of user-related functions. It allows the management of personal data and settings. The access to the personal list is secured by a password. Other creators are not able to change the entries of the personal data.

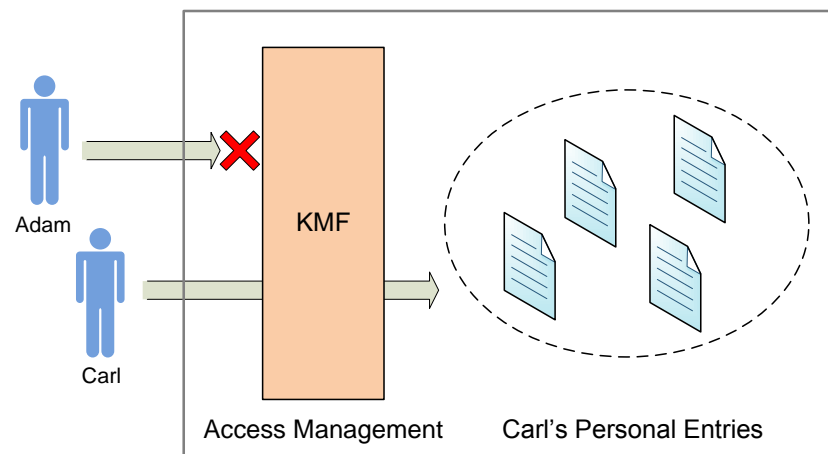


Figure 5.6: Personal Knowledge Access

Figure 5.6 is somewhat similar to the one, describing the basic elements of the knowledge repository. The grey surrounding box represents the knowledge repository. The access management is organised through the KMF and not the front-end software.

All personal data and knowledge is hidden for the other users of the system. A direct exchange of personal behavioural patterns is not supported through the KMF. This could be made available through a front-end application, if required.

5.3 Knowledge Exchange

The exchange of knowledge, in form of behavioural patterns, is a very important feature for the knowledge repository. The following four points describe the main aims of the knowledge exchange.

- Identification of common knowledge.

This enables interesting analysis approaches. It can help to understand what kind of knowledge about the creative process is of interest for the creators. Common knowledge also provides an entry point for beginners.

- Increase of the common knowledge.

The common knowledge is only increased through the exchange of knowledge. A better identification process also increases the common knowledge.

- Improvement of the knowledge quality.

Knowledge entries, created by many are often more precise and less vulnerable to errors. A knowledge entry is revised by its owners before it can become common knowledge.

- Creation of a central knowledge repository.

A knowledge repository is only useful if it stores and enables access to the knowledge of many. The usage of a central instance simplifies the exchange of knowledge.

NB, the term *common* knowledge includes the common and domain-specific knowledge in this case.

As mentioned before, one of the major facilities of the KMF is the possibility to exchange knowledge between creators. The users of the system itself will be mostly unaware of the exchange, because they will not actively distribute the knowledge between each other. Instead, users can choose from a selection of public behavioural patterns. This public knowledge is provided through the domain-specific and common knowledge lists. The reason for choosing an automatical knowledge distribution instead of a manual distribution, initiate by the creator himself, is the so-called information-exchange dilemma. Many studies report that users tend to have a low motivation to share their knowledge with others [72]. This can be very harmful for a knowledge management facility, which is designed for the exchange of knowledge.

Inside the KMF, every creator possesses a personal list of behavioural patterns. The list represents all the knowledge the creator uses for the analysis of his creative process. This list consists of his own patterns and references to selected patterns from his domain or common patterns. Those references are selected from the lists for common and domain-specific knowledge. The user is able to remove entries from his list. However, the personal knowledge entries are not completely removed from the system. The knowledge is only hidden and can always brought back by the user. The domain-specific and common knowledge list entries are completely removed. This will not remove the knowledge entry itself, but only the reference to the knowledge entry.

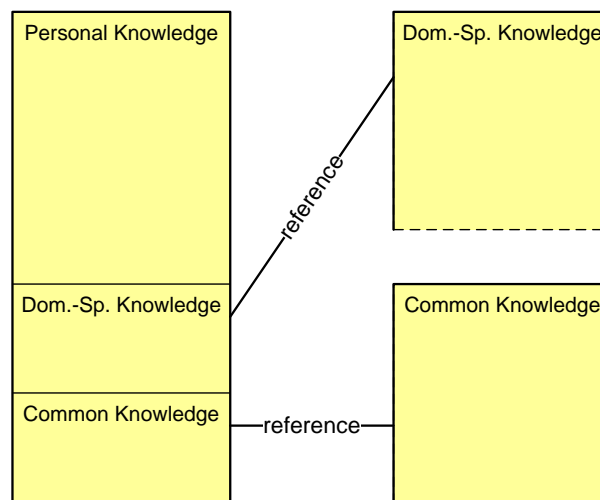


Figure 5.7: Personal Knowledge List

Figure 5.7 depicts a basic example of a personal list of behavioural patterns and thereby represented knowledge. One can see that the entries for the domain-specific and common knowledge contain only a reference to separate lists. These lists avoid redundant entries. The behavioural patterns of the common and domain-specific category are the same for everyone. Hence, there is no need to store these entries in each personal list. This benefits an economical usage of the resources, used by the database. Furthermore, it eases the management of those knowledge entries.

The exchange of knowledge is carried out through the KMF. The process itself is semi-automatic. An identification process will identify possibly identical patterns. But the final decision about adding it to the domain level is made by the owners of the pattern. The hierarchy of the patterns is very important for the exchange. Personal behavioural patterns are only compared with the personal patterns from other creators of the same

domain. The pattern will be taken into account for adding it to the list of domain-specific behavioural patterns, if it is found in a reasonable number of all personal profiles of that domain. The approach for the identification of common patterns is similar. A behavioural pattern is identified as common knowledge, if it exists in several domains. This approach reduces the search space for the identification process. Furthermore, it reduces the occurrence of false positives (matches). Similar expressions for behavioural patterns in the same domain will be much more likely used for the same analysis than in case of e.g. a computer scientist and a novelist.

The knowledge repository provides a special list for users in case one or more of their behavioural patterns have been identified as possible candidates for the domain-specific category. This list enables a discussion whether a knowledge entry of a behavioural pattern is released for the domain. The list contains the meta-information from the creator as well as the meta-information from the other creators with the (possibly) same behavioural pattern. This enables the creator to decide, if he thinks the pattern has the same meaning for the others or not. The KMF provides functions for a vote in this case. The creator can either vote with “*yes*” or “*no*”. If he decides the patterns have not the same meaning, he can vote with no. In contrast, if he thinks it has the same meaning, he votes with yes. The pattern is only added to the domain-specific category, if several of the creators think it has the same meaning. The necessary minimum number of yes-votes is defined by the KMF. The KMF enables the ability for the creators to modify the meta-information knowledge entry, before it is added. This allows them to define a proper set of meta-information for the behavioural pattern.

5.3.1 Search and Comparison

The KMF provides an interface for the search of behavioural patterns via keywords. The functionality enables creators to search for suitable behavioural patterns in the domain-specific and common categories. The result of the search is a list containing a set of related behavioural patterns. The searching process itself compares the provided keyword with the keywords of the meta-information from the knowledge entries of those categories. However, this approach will cause some problems.

Collaboration and knowledge sharing increases the complexity of the knowledge repository in regard to the management and processing of the knowledge flow. The reason for this is the fact that every person uses his own vocabulary for the description and labelling of the activities. It is often the case that people use different words to describe the same

activity. This synonymy of words is a huge problem for the comparison of knowledge and information.

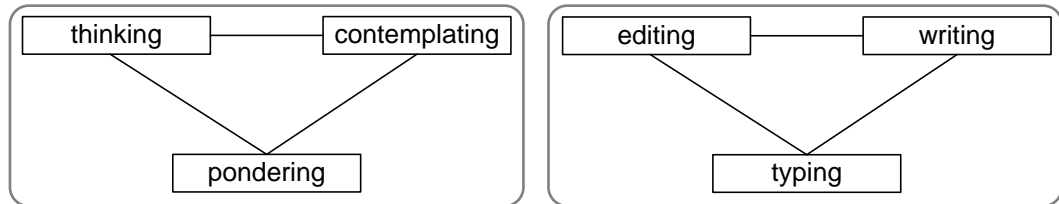


Figure 5.8: Synonymy of Activities

Examples of synonymy are presented in Figure 5.8. The two boxes contain words with a similar meaning. Each word of the left box describes a thinking activity. Those words are slightly different in their meaning. Nonetheless, a creator, searching for patterns related to the *thinking* activity, will be also interested in patterns which are described by *pondering* and *contemplating*. The words in the right box have a similar relation to each other. In fact, there exist related activities for most of the entries in every creator's list of activities.

Another problem arises through the granularity of the list of activities. Some creators just use a few activities for the mapping of the creative process. Others might use a more fine-grained set of activities. Using several activities, where others would use one to describe the same cases. An example is the activity *discussing*. Some people differentiate between relaxed *chatting* with friends and *discussing* with colleagues.

The KMF must be able to provide all closely related entries for a keyword. This requires information about the relations of the words. It is not possible to identify those similarities with only the help of an algorithm. There are several techniques available, which try to solve these problems. Many of them solve the problem of synonymy, but cannot cope with the granularity of the activities. A solution for this problem requires a corpora of words and their relations. This can be either provided by lexical or ontology based approaches or Self-Organizing Map (SOM). This problem has been discussed in detail in Chapter 2.

The KMF uses search facilities based on the Wordnet corpora. This approach has a huge benefit in opposite to the SOM approach. It provides a large set of data. Furthermore, there is no need for a training of the data, before using the system. The SOM approach requires all these steps.

5.4 Summary

This chapter introduced the concept of a knowledge repository. The reason for the development of a knowledge repository arose from the discussion of current creativity support tools in Chapter 2. One of the main limitations of current approaches is the missing of techniques for a store of knowledge. Without this, it is impossible to store and broadly communicate novel findings about the creative process, gained through the usage of the creativity support tool.

The introduced knowledge repository contains a knowledge database and a knowledge management facility. The knowledge database is responsible for the storage of the knowledge itself. Its centralised design enables the exchange of knowledge via networks.

A facility for the management of the knowledge provides interfaces for the adding of new knowledge, change and retrieval of existing knowledge. Moreover, it automatically detects identical patterns within the knowledge entries of creators from the same domain. The most important feature, however, is the exchange of knowledge through shared domain and common knowledge entries. The facility is supporting the exchange and therefore the growth of knowledge. This is an important improvement in regard to current creativity support tools.

Detailed descriptions about the structure and meaning of knowledge for the analysis process are provided. The knowledge, stored in this repository, consists of two types:

- Pattern expression
- Metainformation

Each knowledge entry consists of a BPDFL expression with a set of associated metainformation. The expressions, describing the behavioural patterns are an important part of the knowledge entries. These expressions describe the structure of the wanted information. The purpose of the expressions and the gained result sets is described through the metainformation.

The last section of this chapter discusses the exchange of knowledge in detail. A key focus lies on the benefits which will arise from a centralised knowledge repository with shared data sets. Different solutions for tackling word synonymy have been discussed.

Chapter 6

Analysis and Support of the Creative Process

Objectives

- Analysis of the creative behaviour by using behavioural patterns.
- Introduction of a novel framework for the support of the creative process.
- Discussion of the analysis possibilities.
- Comparison of creativity maps.

“Clearly no group can as an entity create ideas. Only individuals can do this. A group of individuals may, however, stimulate one another in the creation of ideas.”

Estill I. Green

6.1 Introduction

This chapter is answering the research questions number four: “What type of information can be extracted from the creativity maps?” and five: “How can the results of the analyses be used for the support of the creative process?”. The chapter is divided into two main parts. The first part describes the possibilities for analysing the creative behaviour and comparison of creativity maps. The second part describes how the results of the analysis process can benefit the support of the creative process.

The scope of application for the results of the information extraction process will be described in detail. This includes the first step of analysing the extracted information. The creator can undertake the analysis at two stages, during his/her work and more detailed afterwards. Moreover, creativity maps enable the analysis of differences between creativity maps through a comparison. Finally, an introduction of a concept for the support of the creative process which is realised through the analysis of creativity maps with behavioural patterns.

Both the analysis and the support of the creative process are of indirect nature in the here presented framework. The analysis is based on information about the creative process, which is mapped through creativity maps. The results of the analysis are used to stimulate the creativity of the creator by supporting the creative process. Through this, the need to develop techniques for a direct assessment of the creativity can be avoided. Studies about the automatical assessment of creativity have indicated that this process seems to be very complicated, if not impossible. Computational models, focussing on the assessment of creativity itself lack of facilities to properly describe all dimensions of a creative artefact [107].

6.2 Analysis Process

The analysis process is using the information, which was extracted from the creativity maps. This information was extracted through the application of behavioural patterns. The process also requires parts of the meta-information, mainly the description, the type of application and the optional message and constraint, if existing. The optional constraint for the creation of a partial creativity map needs be applied before the information extraction and analysis processes. The application of the behavioural patterns is performed in a sequential manner. A parser instance is handling one behavioural pattern at a time. However, a usage of multiple parser instances is possible.

Creativity maps provide several opportunities for the analysis of the creative process. The analysis process is focussing on the creative behaviour of the creator. The aim of the analysis is the identification of wanted and unwanted behaviour and the creation of statistics. Furthermore, a comparison of creative behaviour in different projects from the same or other creators. Those creators can be from the same domain or even from a non-related domain.

The analysis of the creative process with behavioural patterns knows two different modes: on-the-fly or detailed analysis. The on-the-fly process will constantly analyse the creative process, but only during the working period of the creator. It is furthermore an automatical process to enable a silent monitoring. The detailed analysis, however, is performed as the analysis of the whole creativity map.

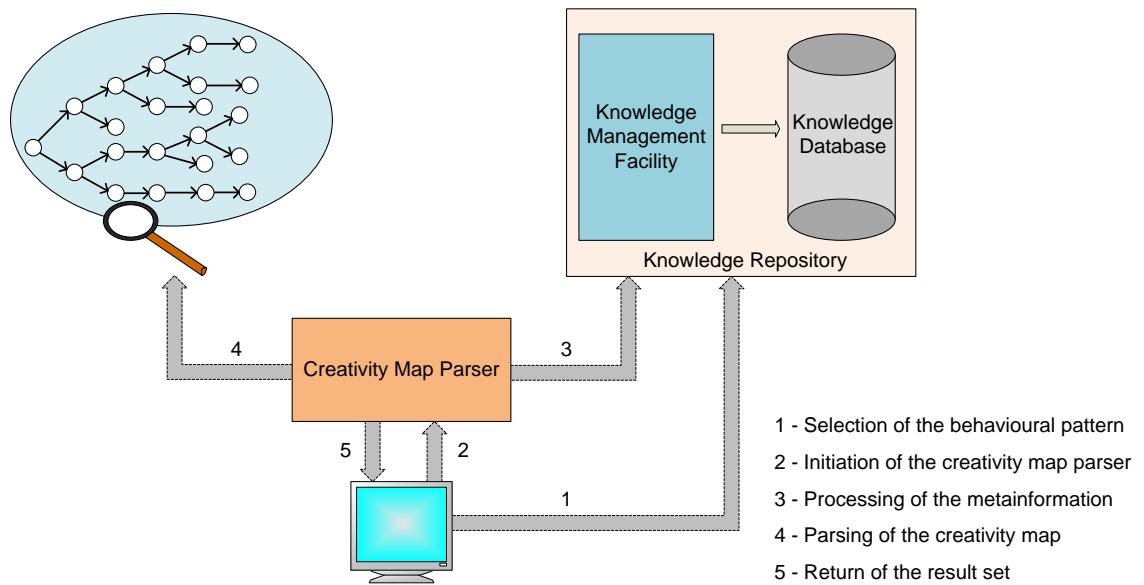


Figure 6.1: Analysis Process

The analysis process, either detailed or monitoring is always initiated by the creator. Figure 6.1 depicts the steps of the process. The user interface for the analysis application is represented by the monitor on the bottom of the figure. The creativity map parser is responsible for reading the BPDFL expression of the behavioural pattern and the extraction of requested information from the creativity map.

The process for the analysis itself is divided into five sub-processes. Step 1 is initiated manually. The creator chooses one or more behavioural patterns for the extraction of information for the analysis. The references of the behavioural patterns will be sent to the creativity map parser in step no 2. The creativity map parser will then look up the entry for the reference in the knowledge repository. It will retrieve the expression for the behavioural pattern as well as check for an optional constraint. The creation of a PCM will be initiated in case there is a constraint. These steps are represented by the third arrow in the figure. The fourth step initiates the parsing process itself. The parser will extract the information which is described through the behavioural pattern. In the last step the result set is returned to the creator. The creator is then able to continue working with those results.

The analysis process is closely connected to the creation and management of knowledge. The input for the analysis often requires a refinement and therefore a change of the behavioural pattern and its metainformation. Hence, it is useful to combine the user interface for the knowledge management facility and the analysis tool. Nonetheless, this is not a requirement. The modular design of the system also allows the usage of different tools for the tasks.

As mentioned before, the creativity map and behavioural patterns enable the analysis of the creative process at two different stages. These stages are listed below:

- Monitoring of the Creative Process
- Detailed Analysis

The scope and differences of these two stages will be described in detail in the following two sub-sections.

6.2.1 Detailed Analysis

The most interesting insights of the creative process are provided through detailed analysis of the creativity maps. This analysis approach enables a thorough inspection of the creative behaviour. It is mainly performed after major contributions to the artefact or towards the end of the project.

The detailed analysis can include the usage of all possible behavioural patterns. The aim of this analysis mode is a detailed overview about the creative behaviour of the

creator. The results are used to gain better understanding about the creative performance of the creator. Through this, it is possible to create new knowledge about the creative process. The knowledge about the creative behaviour and especially the knowledge for the monitoring process is gained through the detailed analysis.

Following is a listing of several, but not all, use cases for the detailed analysis.

Searching for Certain Behaviours

One of the the most important analysis cases is the search for a certain creative behaviour. The behavioural pattern for this case would be a description about the behaviour in the form of a sequence. This pattern might describe wanted or unwanted behaviour. The analysis process allows the creator to take measures in case of the appearance of that behavioural pattern. The parsing process is not performing any kind of similarity measures, as this would lead to false positives. A sequence, describing wanted or unwanted behaviour is always defined precisely.

This type of analysis is the most important one for the monitoring of the creative process, which is described in the next sub-section.

Information about Creational Activities

The analysis of creational activities will provide a lot of information about the working habits of the creator. Examples for this have been presented in Chapter 4. Especially the sequences of activities which lead to the creation activities contain very valuable information. With these sequences, it is possible to compare the creational behaviour e.g. at the start and the end of the project.

Similar analysis approaches with intellectual activities also enable the discovery of important behaviours. This could be for example the percentage distribution of activities as successors of a certain activity. The creation of such statistics is one of the main interests for an analysis.

Analysis of Branching and Merging Operations

The branching operator of the Behavioural Pattern Description Language allows the extraction of branches from a creativity map. A field of application for this operator is the creation of statistics about the successors of an activity type.

The merging operator is more limited and mainly used to create an overview about the predecessors of an activity type.

Information about Revert Operations

The revert operations are useful to identify the reasons why creators stop working on an approach and continue to work on a previous idea. Analysing the circumstances of this phenomenon provides detailed information about possible poor creative behaviour.

A possible indicator for poor creative behaviour is the occurrence of branches with short sequences. This behaviour shows that the creator got stuck with his approach and therefore indicates poor creativity.

Analysis, based on partial creativity maps

It was already mentioned that partial creativity maps provide a lot of opportunities for a refinement of the search space. A PCM allows the application of all above mentioned analyses. Moreover, it enables to include conditions for the states. This approach makes it possible to analyse the influence of the creative behaviour on the artefact. It is now possible to identify the productive phase of the creator, through e.g. a selection of states, representing large changes to the artefact.

A PCM is also useful for hiding viewpoints. Through this, it is possible to analyse the arrangement of activities from certain viewpoints. It is beneficial for some analyses if unnecessary information is hidden.

6.2.2 Monitoring of the Creative Process

Creativity maps are updated in real-time. States and transitions are added to the map at the moment of their occurrence. Hence, it is possible to analyse the current creative

behaviour of a creator. This allows a constant monitoring of the creative process. The aim of this approach is the identification of a certain creative behaviour. The analysis process is running parallel to the creator's working process. It is not interfering with the working process itself and invisible for the creator most of the time.

The monitoring approach is limited in regard to the detailed analysis. A lot types of analysis are not applicable, because these are not of interest for a constant monitoring. For example, the creator would not benefit from a constant identification of all revert activities. The result of the analysis would always contain the same entries, with a minor change from time to time.

Instead, the monitoring process is based on the knowledge about the creative behaviour, which was gained through the detailed analysis process. This has an impact on the nature of the behavioural patterns. The behavioural patterns can be divided into two logical sub-categories:

- Explorative Patterns
- Supportive Patterns

Behavioural patterns of the first category are used to gain new knowledge about the creative process. The aim of applying these behavioural patterns is the collection of information in order to provide an holistic view about the creative performance of a person. It is obvious that most of the behavioural patterns for the detailed analysis belong to this category.

The second category describes those behavioural patterns, which are used for the monitoring process. The task of this process is mostly of supportive nature. The identification of a behavioural pattern, representing poor creative behaviour will not generate a lot new knowledge. The main task of these patterns is the guidance of the user, e.g. to warn him about poor creative behaviour.

The classification of the behavioural patterns into one of these categories is not strict. The usage of a behavioural pattern can change and the categorisation is therefore matter of the creator himself.

The most important task of the monitoring process is obviously the search for poor creative behaviour. Poor creative behaviour can for example lead to poor results at the artefact or might indicate a distracting activity. A creator might know that the creative behaviour in form of the sequence

$$S := \longrightarrow_{reading} ; \longrightarrow_{contemplating} ; \longrightarrow_{writing}$$

has a bad influence on his or her creativity and results in poor contributions on the artefact. The monitoring process is able to search for this sequence with the possibility to inform the creator in case of detecting it.

Another poor behaviour of the creator might be a too long sequence of transitions between two creational activities.

$$S := \underbrace{\longrightarrow_{writing} ; \cdots ; \longrightarrow_{writing}}_{length > 10}$$

The behavioural pattern for the sequence above would contain the LEN()() operator to search for such a sequence with a length longer than ten.

The two examples illustrate that the monitoring process mainly aims on the detection of a certain creative behaviour. Nonetheless, it is still very flexible within this process. The creator itself is able to choose from his set of behavioural patterns. This allows an adjustment of the monitoring process to the current needs of the creator.

The monitoring process reads the transitions one by one and stores them in a cache if the transitions fits the pattern. It starts with the first transition of the behavioural pattern. In case it is found, it checks if the next transition is also the next transition of the behavioural pattern. This is repeated until the pattern is complete or, more likely, the process is aborted as the latest transition does not fit the condition.

A creator might define following behavioural pattern which describes poor creative behaviour:

$$S_b := \longrightarrow_{discussing} ; \longrightarrow_{contemplating} ; \longrightarrow_{discussing} ; \longrightarrow_{prototyping}$$

The current monitored range of transitions contains following activities:

$$S_r := \longrightarrow_{reading} ; \longrightarrow_{writing} ; \longrightarrow_{discussing} ; \longrightarrow_{contemplating} ; \\ \longrightarrow_{discussing}$$

One can see that most of the transitions of the behavioural pattern are found within the current range. However, the last required transition did not appear yet, or will probably

not appear next. It is important to note that only a complete match of the behavioural pattern will trigger an action. An approach to include similar behaviours would lead to many false positives, especially with small behavioural patterns. The strict handling is due the fact that the arrangement of the transitions in a sequence is crucial for the analysis.

The monitoring process often searches for several behavioural patterns at the same time. Each behavioural pattern has its own analysis range and cache. The aim of the monitoring is mostly the prevention of poor creative behaviour. There is no reason to tell somebody that the current approach is a good one. However, there is the possibility to inform a creator that he had already a good idea, but then went wrong somewhere. In this case, the creator might welcome a hint.

Analysis Range

The monitoring process only makes sense for a certain range of transitions. This range is usually a small number of the last added transitions. For example, the process would only analyse the last ten working steps of the creator. An analysis of whole creativity is unnecessary, because the current work will not affect older parts of the creativity map. This benefits to a faster and less complex analysis process.

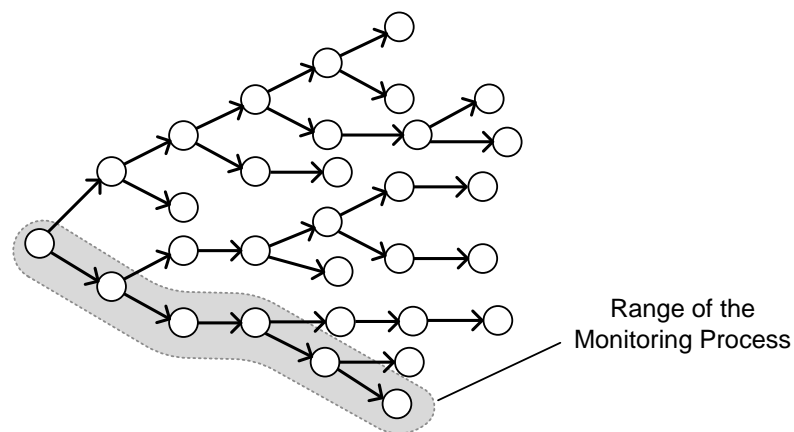


Figure 6.2: Monitoring of Current Behaviour

The Figure 6.2 depicts a range for the monitoring process within a creativity map. The range is not always covering the sequence from the initial state of the creativity map to the most recent state. The range often contains only transitions for a sub-sequence of the

before mentioned sequence. This emerges due to the knowledge of previous appearances of the same behavioural pattern.

The range changes with every new transition. This requires a sophisticated processing of the activities as one new transition might lead to an identification of a behavioural pattern. The monitoring process uses a cache in order to enable a quick analysis of the range with every new transition. The range itself is not a static number of transitions, but the maximal amount of transitions for the behavioural pattern.

6.2.3 Analysis of Frequent Behaviours

The process of extracting information from the raw data of the mapping process is not limited to the creation of creativity maps. A second feature is the identification of frequent behaviours. A frequent behaviour is a sequence which often reoccurs inside the creativity map. Hence, it is a special behaviour, providing information about the habits of the creator. A frequent behaviour is always a sequence without branches. The composition of frequent behaviours starts with the identification of the minimal frequent sequences, which are then combined to find larger frequent sequences. The analysis of this data is very interesting, as it is established that the sequences are frequent and hence very valuable for the search of poor or good creative behaviour in it.

An introduction of the techniques for the identification process itself is not part of this thesis. Hence, it is assumed at this point, that the technology for the identification process is already available. The examples of frequent behaviours have already been identified through a preceding process. The approach is not using any type of knowledge for detection process, in contrast to the here presented techniques. However, a short explanation about the creation of the frequent behaviours will be provided, in order to enable an understanding about the structure of the retrieved information.

The identification of frequent behaviours starts with a search for sequences with a length of two transitions, which often occur within the creativity map. The frequent behaviours are stored in a buffer for a second processing step. The second step aims for combining sequences into longer sequences. In other words, two small frequent behaviours will be glued together in order to get one large frequent behaviour.

The following three examples of frequent behaviours represent the output of the identification process within the creativity map. The frequent behaviours FB_1 and FB_2 were identified during the first step of the process.

$$FB_1 := \longrightarrow_{reading} ; \longrightarrow_{contemplating}$$

$$FB_2 := \longrightarrow_{contemplating} ; \longrightarrow_{writing}$$

The *contemplating* transitions in those two behaviours describe the same transition in the creativity map. Hence, the start of the transitions is the same $St(FB_1_contemplating) = St(FB_2_contemplating)$ as well as the end of the transitions $Et(FB_1_contemplating) = Et(FB_2_contemplating)$. Therefore, it is possible to combine the two behaviours. The result of combining the two frequent behaviours is represented by the frequent behaviour FB_3 .

$$FB_3 := \longrightarrow_{reading} ; \longrightarrow_{contemplating} ; \longrightarrow_{writing}$$

The newly created frequent behaviour consists of three transitions, namely the *reading* transition of the first behaviour and the *writing* transition of the second behaviour, merged through the *contemplating* transition. The two initial frequent behaviours will be dropped out of the buffer and replaced by the merged behaviour.

Frequent behaviours which could not be combined with others will be removed from the buffer.

During the third step, the identification process will again search for merging candidates within the newly created frequent behaviours of step two. This process is repeated until it cannot merge any more of the frequent behaviours. The aim of the merging process is the identification of the maximum frequent behaviours. The longer the behaviours become, the more analysis types are possible. The identification process must also take measures that the long sequences of the behaviours are still “frequent”. One or two appearances of a frequent behaviour would not justify its name.

The identification of frequent behaviours is not realised through a constant monitoring process. It is only useful for the detailed analysis. Nonetheless, it is a very important part of it. The frequent behaviours are ideal for an analysis of their influence on the creative process. It is comprehensible that a frequent behaviour might help to create a behavioural pattern for the monitoring process, depending on whether the creator categorised its influence as positive or negative.

6.3 Hopping

The creativity of a person is hugely influenced by the work of other people. An idea is never completely novel and independent from the work of others. Creativity describes the ability to arrange and combine already existing ideas and approaches in a novel way. A mathematician is not able to make a discovery without building up on already existing ideas. Musicians are always inspired by the works of other creators from the same or also other domains. Moreover, it is not only the idea of others which make people creative, but also the interaction with those creators. Interaction supports the exchange of knowledge.

A single creativity map provides a lot of opportunities for analysing the creative behaviour of a creator. It allows the analysis on various levels, e.g. depending on viewpoints or activities. After the interpretation of the collected information, it is certainly of interest to compare the creativity map with creativity maps of the same or other creators. This is beneficial especially for creators, who work collaboratively in teams.

The comparison of creativity maps enables the search for similarities and differences in the creative behaviour of creators. Furthermore, it is possible to compare the creativity maps of creators from other domains. This will support the analysis of creativity in general and might solve the questions whether there are general patterns of creative behaviour, occurring in creativity maps of all domains.

One important phenomenon of creative behaviour is the “*hopping*” between domains. It describes the interaction of creators, possibly working on different domains. The analysis of this phenomenon is very important as creativity often occurs where the ideas of people from the same and other domains are combined. It is obvious that the hopping phenomenon is crucial for creation of something new. Creativity usually appears on the boundaries (edges) of two or more domains.

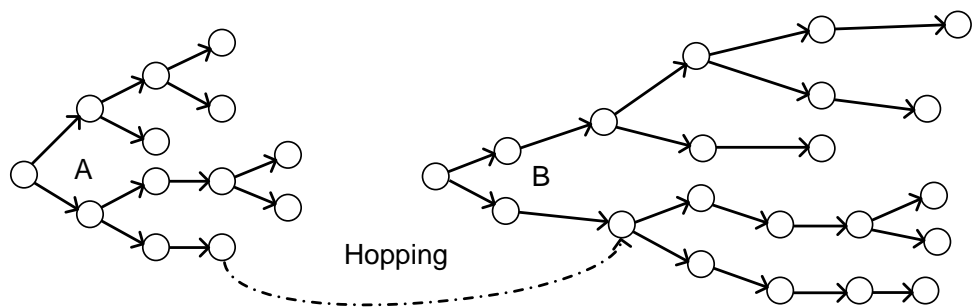


Figure 6.3: Hopping between Creativity Maps

In Figure 6.3, the creativity map A could belong to the project of a musician and the creativity map B to a writer. In such a case, the musician might have read an inspiring novel or talked to the novelist about his work and got inspired by that. The link between these two creativity maps would be described as hopping. A hopping within the same creativity map would be nothing more than a revert. However, it would not share the same meaning. Furthermore, the hopping is not a real operation. The creator would not contribute to the creativity map of the other creator, he got inspired by. It only describes a relation to a specific point in the artefact of somebody else.

Hopping is only possible between states of different creativity maps. A hopping between transitions of different creativity maps is impossible. The transitions only represent the activities and not information about the artefact itself. The state, however, contains the valuable data and information for the creator.

The analysis of the hopping is quite difficult, if the creator did not save any information about this event. Nonetheless, the creativity maps enable to gain some information about this event through the above mentioned comparison with other creativity maps. Another, maybe even more valuable approach for this event might be the evaluation of collaboration.

6.4 Comparing Creativity Maps

The big advantage of the in this thesis presented approach is the possibility of comparing two or more creativity maps which each other. The foundation and rules are the same for all creators, independently in which domain it is used. The only, but important difference are the arrangement of its components. There will be never two identical creativity maps. Not of two different creators or even two identical created by the same person. The different compositions of the creativity maps are very valuable in order to compare the creative process of two creators. The results from analysing and comparing the creative processes of creators might allow to better understand creativity itself. The comparison of creativity maps allows setting the creative behaviour of one creator in context to others.

Creativity maps enable to identify the strengths and weaknesses of a creator. The analysis might help him to improve his creative abilities, either through direct support with software tools or indirect through a careful study of the analysis results. The results of the analysis are not only beneficial for the creator himself, but for many others of the same domain or even other domains. Creators are able to exchange their experiences and knowledge about their creative process.

The comparison of creativity maps is as important as the analysis of the own creative processes. Both feature a growth of knowledge about the creative process and creativity. However, the comparison is able to present a more general overview about the creative process. The information retrieval is the same in both cases and based on the usage of behavioural patterns. Moreover, the two tasks are not isolated from each other. The knowledge about the own creative process will help to find similarities in the comparison step. The analysis of the own creativity maps, on the other hand, will also benefit from the knowledge gained through the comparisons.

The use of such comparisons is very helpful for teaching purposes. Through this, pupils and students are able to analyse their behaviour when solving a problem. The analysis of their creative behaviour enables them to compare their creative processes, which is also the learning process in this case. The teacher will be able to check the progress of the pupils by comparing the creativity maps with the maps of pupils from the years before. The analysis also enables the comparison of different teaching approaches. The effects of a teaching approach A on the creative process can be compared with the effects of a second teaching approach B. Studies about the impact of a teaching approach on the creativity have already been undertaken [3]. However, the structure of the creativity maps allow to make meaningful assumptions on why an approach is more favourable than another one.

There exists always a possibility of negative effects when trying to use the working methods of other creators. This cannot be avoided and will happen from time to time. Nonetheless, the creativity maps cater for solving these negative effects. A later analysis of the creative process would reveal the problems, if the creator would not recognise the, for him unsuitable, approach.

The comparison of creativity maps from two different creators often leads to the problem of synonymy of activities. The evaluation of the analysis requires the comparison of the same activities. Otherwise, the results will be meaningless. This problem was already addressed in Chapter 5. The knowledge management facility provides functionalities to solve problems with synonymy. Nonetheless, this functionality cannot support the comparison of creativity maps of creators from different domains. The person who initiates the comparison must identify the equivalent activities of both creativity maps. For example, somebody wants to compare the creativity maps of a computer scientist and a creative writer. The creational activity for the computer scientist would be *coding*. However, the creational activity for the creative writer would be *writing*.

The reason the KMF cannot cope with this categorisation lies in the fact that the computer scientist will most likely also have an activity like *writing* or *noting*. These activities however will not change the artefact and therefore belong to the category of intellectual activities. The meaning of the *writing* activity of the computer scientist is different to the *writing* activity of the creative writer.

One can see that the analysis of the creative behaviour still requires the assistance of a person. The simplicity of the analysis process on the other hand makes it very easy to adapt its components to new issues.

6.4.1 Automated Comparisons of Creative Behaviour

The knowledge entries of the KMF are the ideal resource for a comparison of the creative behaviour. It is possible to use the supportive behavioural patterns for this task. A supportive behavioural pattern characterises a specific creative behaviour of a person. This will mostly be the description of poor creative behaviour.

The usage of this knowledge has a big advantage in regard to the comparison of creativity maps. The approach does not require any analysis of the creativity map itself, as this step has been already undertaken in the past. The knowledge can be directly retrieved from the database. The comparison of the entries is based on their metainformation. This enables a fully automated comparison process. The access to a knowledge entry requires of course the approval of its owner for privacy reasons.

6.5 Analysis of Collaboration

Collaboration or co-creation is a very important part of our daily working life. Today, most people work together cooperatively on projects. Technical projects like software development or engineering often require teamwork, if the growing complexity exceeds the capabilities of a single person. New technologies have increased the amount of collaborative art and work. Especially the internet enables new ways of creative artwork. Through the cross-linking nature of the internet, this artwork is often created through collaboration.

A different type of collaboration becomes more and more popular as well - collaborative learning. There is an ongoing intensification of collaborative learning on the university and school sector [98]. Especially on the domain of Computer Supported Collaborative Learn-

ing (CSCL). The wide availability of internet connections enables workplace-independent collaborative learning. It is certainly of interest to analyse how this different learning approaches influence the learning abilities and the for this necessary creativity.

The analysis of collaboration aims for different objectives than the comparison of creativity maps. The aim of this type of analysis is the search for changes in creative behaviour when working collaboratively. It is important to answer the question of what effects collaboration has on the creative process. A creator, working alone on a project, can freely perform his tasks without the need to make arrangements with others about the division of work. This is impossible when working in a group. In most groups, each creator has to follow a work plan in order to avoid interference with the work of others. These limitations will have an effect on the creative process.

Another field of interest for this analysis is the question whether collaboration hinders the creative capabilities of a creator or enriches them. It was mentioned that creativity appears, where many ideas come together. Nevertheless, collaboration will not necessarily bring together the ideas, but the manpower. Hence, it is not sure if collaboration will always benefit the creative process. On the other hand, many of the most recognised researchers and creators worked collaboratively during most of their career [65]. Therefore, the answer will be certainly different for each person. The analysis, however, can help the creator to identify the own weaknesses and strengths when working collaboratively. Through this, the creator has a good chance to improve skills.

The transitions inside the creativity map provide detailed information about the changes of creative behaviour. The changes will mainly affect the communication and interaction of the creators. The timestamps inside the states help to compare the activities of the project member for a certain time. A certain period of time can be analysed through the usage of PCMs. Through this it is possible to investigate the interleaving of activities performed by several users.

6.6 Creativity in the Digital and Non-Digital Age

The digital age changed the way of how people communicate and interact. The internet and mobile phones have increased and simplified communication. Furthermore, today it is possible to communicate in groups rather than one to one. The speed of delivering messages increased dramatically. Where a letter took several days for the delivery, an email only requires seconds to be send to the other side of the world. This influences the

way how people correspond with each other. Most people will think carefully about the text for a reply to a letter. It can take days before somebody found the, for him adequate, text for the response. An email, however, is usually replied quickly. The composition of the text often takes only minutes. The reply to a short message on a mobile phone often only seconds. These changes of the communication habits must affect the creative process.

Another novelty of the digital age is the way how new pieces of art are created. Today, musicians use computers for the creation of new songs. Movies and tv series are at least processed with computers at some stage of the creation. Much of the current content is partially based on computer generated material. There are also fully on computer technology based movies and cartoons.

“The computerization of culture not only leads to the emergence of new cultural forms such as computer games and virtual worlds; it redefines existing ones such as photography and cinema.” Lev Manovich [84]

The usage of computer technology influences the creative process. On the one hand it can hinder creativity where creators cannot freely handle with their creative artefact. On the other hand it can enable new forms of art and therefore inspire and enhance the creativity of people. The introduction of computer technology is often not aiming on achieving more creative products, but reducing the costs of the development process [126]. It is obviously important to analyse whether computer technology benefits the creativity or hinders it. This varies from case to case and needs to be analysed by the creator himself. It is impossible to give a general answer to this question.

The analysis of the influence of the digital technology on the creative process basically focusses on the communication and creational activities. It is difficult to generate a creativity map after an artefact has been created. However some artist left behind a huge collection of information about their creative process. A famous example for this are the notices of Beethoven. Moreover, creators often preserved their communication with other creators - especially in domains like maths, philosophy or natural sciences. Through this, it might be possible to map the creative process of already deceased artists. Naturally, the information content of this type of creativity map cannot be compared to those, created through the creator himself. The creativity maps, based on notices and correspondence of a creator will lack of precision as not all activities have been recorded. Nonetheless, it will provide a lot of important information about the creative process.

6.7 Evaluation of Creative Behaviours

The analysis of the creative behaviour is not only of informative nature. It stimulates creators to gain new knowledge about their creative processes and creative behaviours. It was mentioned before that knowledge is always based on the experience of former projects and the experience of others. The knowledge leads to the creation of new behavioural patterns and therefore to more refined analyses. The evaluation of the results from the analysis process allows to put the results into a context.

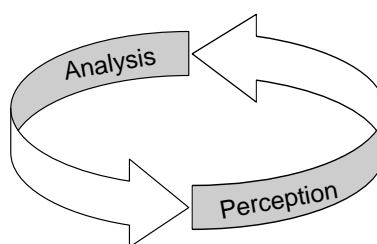


Figure 6.4: Cycle of Analysis and Perception

The interaction of analysis and perception is best represented by a cycle, as depicted in Figure 6.4. Analyses lead to better perceptions of the creative process. The better perception then leads again to refined analysis approaches. Therefore, it is clear that one cannot separate these two stages from each other.

The evaluation stage aims for the appropriate representation of the analysis results. A comparison of creative behaviour requires a format which enables to illustrate the meaning of the results. The result set of the information extraction process provides already all required information about the creative behaviours of the creator. Nonetheless, the information is not presented in the appropriate way. The information of the result set is predestinated to be published as statistics.

6.7.1 Statistics and Probabilities

A very interesting part of the analysis is the creation of statistics. The usage of statistics is ideal for providing an overview about the distribution of transitions and sequences. The data for this statistics is created by the creativity map parser during the information extraction process.

A good example for the creation of statistics is the behavioural pattern which collects the direct successors of a creational activity. This pattern was already introduced in Chapter 4. The behavioural pattern is expressed with the BPDFL as follows:

$\langle pattern \rangle :=$

$SUM(\langle c_transition \rangle \sim \langle a_transition \rangle);$

Assuming, the creator is a writer, the creational activity is *writing*:

$\langle c_transition \rangle = \text{“}writing\text{”}$

The behavioural pattern is collecting all transitions, which directly follow after a *writing* transition. The expression contains the branching operator in combination with the SUM() operator. The result of the parsing process is one element.

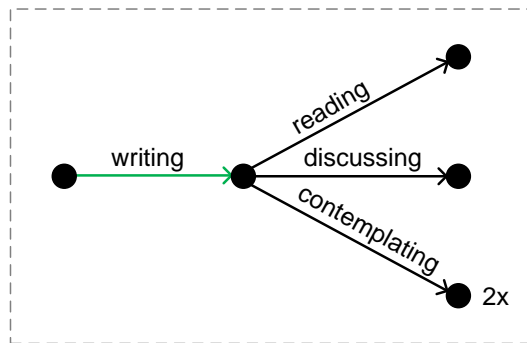


Figure 6.5: Result of the Extraction Process

Figure 6.5 depicts the result of the parsing process. It was mentioned before that the parser will provide the number of appearances for duplicates. A similar function is provided when using the SUM() operator. In this case, the parser will not count the number of appearances for the pattern, because this would always be one, but the number of appearances for every sequence.

The parsing process resulted in the knowledge that there are three activities types which the creator will perform after the *writing* activity. These activities are: *reading*, *discussing* and *contemplating*.

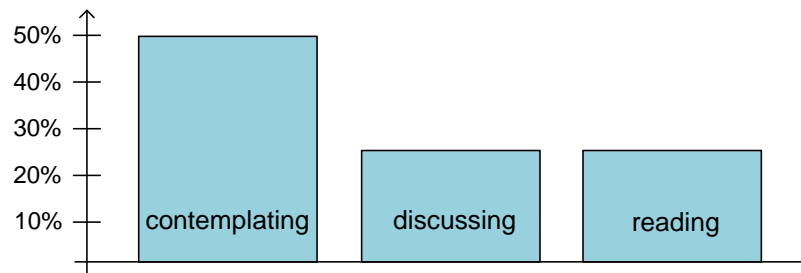


Figure 6.6: Probability of choosing a Transition

The bar chart in Figure 6.6 illustrates the percentage distribution of the transitions. One can see that the creator will most likely perform a *contemplating*, with 50% probability. The *discussing* and *writing* activities will only be performed each in 25% of all cases.

The creator could now further analyse the creativity map to find out which of the options is more likely to let him produce better results and even more important, in which cases the option will lead to poor results. Through this, the creator is able to create new patterns for the monitoring process. The new pattern could inform the creator in case it detects poor creative behaviour through the identification of the behavioural pattern.

6.8 Implications of the Analysis and Evaluation

So far, the focus of this chapter has been on the description of the different possibilities for analysing the creative process. The presented techniques enable a detailed understanding of the creative behaviour of the creator. The benefits for the creator and other interested people are evident. Nonetheless, the real benefit for the creator are the conclusions, which will be derived from the gained information.

The creative product is the result of the creative process of one or more creators. The creativity itself cannot be represented by a single action like e.g. an inspiration. Creativity always emerges through a concatenation of actions. The aim of the detailed analysis is the identification of these concatenations. The in Chapter 4 introduced behavioural patterns do not only enable the identification, but also enable the description of these concatenations. In the latter case, the behavioural patterns will reflect the creativity of the creator. The importance of this fact should not be underestimated. Knowing what stimulates the creativity of a person enables the tailored support of creativity, based on knowledge and not based on assumptions as done in other approaches.

6.9 Creativity Support

The in this thesis presented approach for enhancing the creative process and the resulting creativity differs from the current approaches. The current approaches for creativity support tools were presented in Chapter 2. The weaknesses of those approaches were explained in detail. This section will not introduce one feature, like a graphical stimulus, like the approaches described in Chapter 2. Instead, this section will explain the role of the analysis, knowledge and the application of both for a support of the human creativity. Through this, a universal framework without the limitations of previous approaches will be provided.

The support of creativity is rather a matter of delicacy. It is very much influenced by the knowledge of the creator about his own creative process and his creative behaviours. The knowledge is gained through the exchange with other creators, but also through the analysis performed on his own. The decision about the usefulness of certain information about the creative behaviour is made by the creator and/or the members of the domain. A domain and user driven creativity support is more suitable for catering the requirements of the tasks in that domain. Through this, limitations regarding the possible methods for enhancing creativity are avoided.

A tool-set for the support of creativity must be able to guide the user into directions which enhance the creativity. On the other hand, the tool must not under any circumstances bother the user in his creative process. This would only result into a rejection of the tool-set. However, a creativity support tool is only the one side of supporting the creativity. The creator will gain a lot of knowledge about his creativity through the analysis of his creative process. The in this chapter introduced analysis processes provide the creator with detailed information about the creative process.

The advantage of the presented approach is the focus on the creative process, instead of only analysing the results of the creation process itself. The support is aiming on an improvement of the creative behaviour and not a direct influence on the artefact. The improvement of the creativity and therefore the resulting artefact is of indirect nature. The here presented approach is designed to facilitate a growing of the knowledge database. Existing creativity support tools focus on a specific approach for a stimulus of the creator or even the creation of creative artefact. This often limits the field of application to a small domain. The DMCA caters for the collection of knowledge about the creative process from all of its users. Each contribution of a user will increase the knowledge about the creative process and creativity in general.

Creativity Support is also provided through the monitoring of current creative processes. The monitoring process is designed to initiate external stimuli for exploration and curiosity. For example, if it is known that reading a book or having a discussion has a positive effect when the creator is struggling with his work.

The monitoring is especially favourable for teaching purposes in schools. The students will receive an immediate feedback after a certain behaviour was detected. This behaviour would most likely indicate that the student is making a mistake. The problem of a too strict teaching philosophy through a general, but unsuitable behavioural pattern is not given. Each creator, or student in this case, has a set of personal behavioural patterns, which overrule other behavioural patterns, provided by the domain.

The difference between the DMCA and other projects is the fact that the here presented approach is not only focussing on providing a certain stimulus, but instead providing stimuli, guidance, exploration and understanding of the creative process and behaviour of creators. Only an open approach, like this, is capable of satisfying the needs of a creative mind.

Measuring the effectiveness of a creativity support tool is very difficult. There is no general taxonomy for the evaluation of creativity. The identification of such a taxonomy would mean to have a general definition of creativity itself. This is of course not possible.

Nonetheless, the here presented approach enables to analyse the creative process of a creator over a long time period. The analysis of the creativity maps will show whether previous identified poor creative behaviour continued to re-appear after technologies like the monitoring process were used to overcome the problem. Hence, it is possible to analyse the influence and therefore the impact of certain approaches for improving the creativity through the effects those approaches had on the creative process. Changes in the creativity map itself will show if the creativity has improved.

6.10 Summary

This chapter provides a detailed introduction into the analysis of creative processes. A broad range of possible analysis types and approaches were discussed. It was explained that the combination of a pattern description language, the concept of behavioural patterns as well as a knowledge management facility enable a powerful and flexible analysis of the creative process.

The strength of this approach is the internal structure of the creativity maps. Every creativity map is based on the same foundation. This allows the comparison of creativity maps and as a result the comparison of the creative processes of two or more creators. It was explained why collaboration and hopping plays an important role. Current Creativity Support Tools are not able to cover the aspects of hopping at all. Hence, the approaches do not allow to study a very important phenomenon: Creativity often occurs on the boundaries between domains (see Chapter 2).

The structure of the creativity map enables not only the analysis of the creative process, but also to study its influences on the artefact, which is stored inside the states. It is possible to either include the states into the analysis process by using partial creativity maps or by manually examine a selected version of the artefact.

The reader has learned that presented approach enables more than one analysis type, namely the detailed analysis, the monitoring process and the analysis of frequent behaviours. It was explained that the detailed analysis is used to gain new knowledge about the creative behaviour of a creator. The evaluation of this knowledge is leading to the definition of new behavioural patterns, e.g. describing wanted or unwanted creative behaviour. It therefore provides an indirect creativity support.

The monitoring process uses behavioural patterns in order to monitor the current working process of a creator. Through this, it is possible to identify poor creative behaviour or, in opposite, wanted behaviours at the moment of the occurrence. This type of analysis enables a direct support of the creative process. It was explained that the processes for the analysis and support of the creative process or creativity are mostly overlapping.

A very interesting feature for mapping human creativity is the possibility to produce creativity maps a posteriori. This is possible in case the creator has left behind information about the creation process in form of notes or voice recordings and so on. So far, the creation process, undertaken in the past, could only be examined by hand. A creativity map, based on such information allows studying the creative process with the aid of computer. Of course, it also enables a precise comparison of the creator's creative process with those of others.

The presented analysis approaches reach far beyond current techniques. Moreover, there is no timely limitation for the analysis. Everything can be repeated at a later time as all information and knowledge is either stored in the creativity maps or the knowledge repository. The analysis is not limited to a specific type. Behavioural patterns provide full flexibility for the description of constructs inside the creativity map.

Chapter 7

Tool Support

Objectives

- Presentation of an assistive collaborative framework.
- Introduction of the technique for tracking revert operations.
- Techniques used for the realisation of the BPDFL parser.
- Description of the monitoring process.
- Presentation of a creative working environment and its main tools.

“We’re entering an era in which software learns from its users and all of the users are connected.”

Tim O’Reilly (*1954)

7.1 Introduction

A creative person requires a set of powerful tools and a creative environment for his work. The type of these tools depends on the requirements of the creator. These requirements are very important and need to be taken into account when developing software which caters for the creativity of people. Computer programs are often developed for a very specific task. A program to support the creativity of creators from many disciplines must not share this limitation. Furthermore, it requires a pluggable design for the ability to add and remove tools, depending on the needs of the user.

The De Montfort Creativity Assistant (DMCA) is a tool set, providing facilities to work on a creative artefact and to capture and analyse the creative behaviour of a person during his work. It enables people to work collaboratively on a project. The tool set is based on a client-server architecture. It contains two major client-side applications:

- De Montfort Creative Environment (DMCE)
- De Montfort Creativity Mapper (DMCM)

One of the main tasks of the server-side is the retrieval and preparation of information for the analysis of the creative process. Therefore, this chapter will provide detailed information about the techniques, which are used for the management and processing of knowledge and information. The presented framework demonstrates the applicability of the approaches, introduced in the previous chapters.

This chapter is divided into three major parts. The first part will describe the role of the DMCA and provides an overview about its components. The second part introduces the tracking of the revert operations, which is required for the analysis. This is followed by an introduction of the techniques for the extraction of information from creativity maps. It describes how the expression of a behavioural pattern is for the application on a creativity map. Furthermore, the monitoring process is explained in detail. The third part provides an overview about the tools of the DMCE and the DMCM. A knowledge management and analysis tool, as part of the DMCM, provides a GUI for the in this thesis introduced approaches and techniques.

The DMCM has a modular design, which allows using it as a standalone application. Nonetheless, the aim is a seamless integration into the DMCA.

This chapter provides the solutions for research questions number two, three and five.

The DMCA as well as the DMCE and DMCM have been developed in collaboration by Sascha Westendorf and the author of this thesis, as part of the studies on the Creative Technologies Research Programme at the Software Technology Research Laboratory (STRL), De Montfort University. Hence the author wants to acknowledge the contributions of Sascha Westendorf for co-developing the technologies, described in Section 7.2, 7.6 and 7.7 [22, 131].

7.2 De Montfort Creativity Assistant

Computer-based mapping of human creativity requires a powerful framework with a flexible set of applications, catering for the needs of the creator. The DMCA is a framework with a set of tools, enabling the creation of a creative artefact with a simultaneous capturing of the creative process. Its pluggable design allows the adjustment of the tool set to the requirements of the user.

The DMCA is a web-enabled tool set with groupware-like facilities, aiming at the collaboration between creators. It is encouraging the users to work together on the same product as well as to exchange their knowledge.

The architecture of the DMCA follows a number of principles, which makes it to a universal creativity support tool. The principles evolved from the findings of the literature review in Chapter 2. The four main principles are the following:

Web-Enabled

A web-enabled application must cater for requests of services and resources via a network connection. This includes mainly the Hypertext Transfer Protocol (HTTP) and the Transmission Control Protocol (TCP)/Internet Protocol (IP). Access to the resources is granted or rejected based on a proof of sufficient access rights. The requests might be made by a web browser or through inter-process communication via programming languages like *Java* and *C++*. The requested data or requested operation must be processed in real-time.

The De Montfort Creativity Assistant is a web-enabled application, which can be downloaded and initiated by any browser from any standard website. This gives it the added benefit of allowing it to be platform- and workplace-independent.

Data-Centric

The most important element for creativity support tools is the collection, analysis and distribution of data. The data about the creative process as well as the data about the creative artefact are first-class citizens. Traditional creativity support tools aim for improved design principles in form of processes, implemented as methods or objects. Data is only seen as a second-class citizen.

The here proposed methodology focuses on the data that is moving and transforming in the system, rather than the processes that are performing those actions. Hence, a data-centric approach provides a natural focus on the processes behind creativity rather than on the design of the tools.

The De Montfort Creativity Assistant realises the data-centric approach through following principles [131]:

- Modularisation and loose coupling of code. This is achieved through well-defined interfaces and abstraction.
- Clear separation between data-handling and data-processing components. Data-processing components are likely to be stable while data-handling allows for differing data formats, persistence, and timeliness.
- Code is hidden, but data is allowed to be shared across the entire distributed application, as well as the ability to modify and enhanced the application during its life cycle.
- Data is exposed and is visible to the entire system.

Collaborative

The role of collaboration is very important for the creativity of creators. On the one hand, many of them would not be able to create something special without the interaction with others. On the other hand, some projects require the work force of many people to enable a complete in a reasonable time. Therefore, it is very important that a creativity support tool will foster co-operation and collaboration between creators. The collaboration can take place in different ways. For example, it can be the sharing of ideas through direct communication. Another example is a real-time collaborative environment. It allows sharing ideas, techniques and innovation.

The DMCA provides an integrated platform that enhances cooperation and enriches collaboration in real-time. Facilities such as chat, white-board and pend-it note are all provided in an integrated fashion. It allows a seamless integration of additional applications if required.

Discipline-Independent

It is well recognised that creativity appears where creators cross boundaries of different disciplines. A platform for the analysis and support of the creative process must enable an abstraction of data. This can be only guaranteed through separation between data and information about the creative process and the artefact itself.

The presented approach caters for this necessity. This benefits the integration and removal of tools from the DMCA

7.2.1 Layers of the DMCA

In Figure 7.1 one can see the various layers of the system.

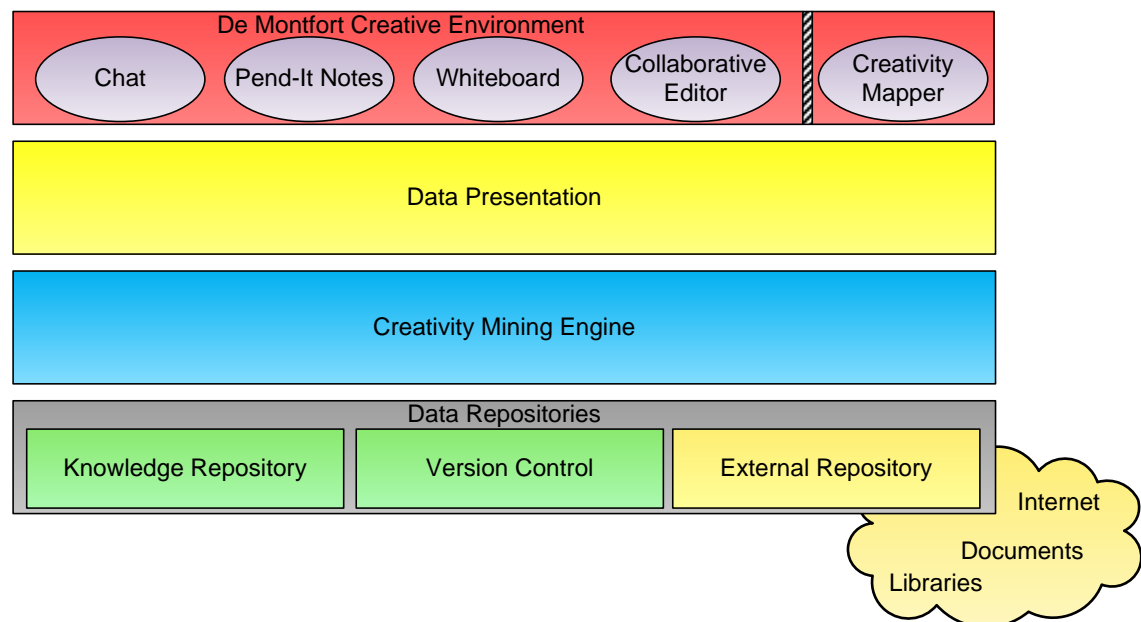


Figure 7.1: Layers of the DMCA

The top-most layer represents the client side of the DMCA. The three layers below are located on the server side. The cloud in the figure illustrates that the DMCA is not a closed system, but using external applications and data sources. The term “External” does not indicate that it is stored on another server. It only indicates that the data was not created through the DMCA and its applications.

The design of the DMCA is based on a multi-tier architecture. The system basically implements a three-tier architecture, but with an additional tier, representing the applications for the user. Each tier or layer has a specific task and provides a certain functionality. The functionality is provided by a set of tools which process or represent the data, involved in the creative process and the creation of an artefact.

The following paragraphs contain a detailed explanation of the tasks for each layer.

De Montfort Creative Environment and Creativity Mapper

The top-most layer represents the GUI of the DMCA. It contains a set of tools, which are used for the creation of the artefact as well as for communication with other creators and other helper tools.

Currently, the DMCE contains three applications: a chat facility with an integrated whiteboard, a collaborative editor for creative writing and notes with a reminder functionality. The applications are managed by the project management facility of the DMCE.

The pluggable design of the DMCA enables to change the main application in order to cater for the needs of the creator. The current version of the DMCE contains a collaborative editor, specially designed for creative writers. The removal of this application and adding another is possible with only minor changes to the system itself.

The De Montfort Creativity Mapper is positioned separately from the DMCE. It also belongs to the same layer - the GUI. However, the application can be used as standalone.

Data Presentation

The data presentation layer is responsible for the transformation of results for a user request into a proper structure. Furthermore, it will transform the input of the user into a structure, which enables further processing. Hence, the data presentation layer provides bidirectional data transformation.

This layer plays an important role for the flexibility of DMCA. Only structured data and well defined interfaces allow a clean integration of additional tools into the system.

Creativity Mining Engine

The creativity mining engine caters for the central analysis and data processing. It contains the methods for the creation of the creativity map as well as the BPDFL parser for the identification of behavioural patterns. The latter technology is described in detail in the following sections.

Like the other layers, the creativity mining engine consists of several individual elements. The facility for the creation of the creativity maps is independent from the facility for the identification of behavioural patterns. This benefits the pluggable design and eases making changes to the system.

Data Repositories

The data repositories represent the databases of the DMCA. Those databases are knowledge databases, a version control system (storing the artefact) or even external repositories like Wikipedia.

- **Knowledge Repository**

This represents the in Chapter 5 described knowledge repository. It contains the KMF and the knowledge database.

- **Version Control**

The DMCA stores every version of the artefact. This version control system keeps track of the versions and the differences between them.

- **External Repository**

This repository is a complementary knowledge source. It enables to access sources like Wikipedia or provides access to the Wordnet database. The word “external” does not indicate the location of the database. It only indicates that the content of the database was not created within the DMCA.

7.3 Revert Tracking

The in Chapter 4 introduced revert operator and the resulting analysis opportunities promise a better understanding about drawbacks and also inspiration, which occurred during the creative process. In that chapter, it was mentioned that the revert operations inside the creativity map cannot be identified through a normal BPDFL expression without the revert operator. The revert operator, however, requires the information about the involved transitions. Therefore, it is necessary to identify the transitions which finally lead to the revert to an older state. The tracking of the revert operations is not undertaken during the analysis process. Such an approach would slow down the analysis process. Instead, the information is collected during the creation of the creativity map. This information is then stored for later following analyses.

The identification of the transitions, responsible for a revert operation is not always a simple task. Some reverts will be applied on sequences, which have already been influenced by another revert operation. Difficulties of understanding can occur when reverting to other branches or nesting. The identification process requires an approach which can cater for all cases. The following example should illustrate the complexity of this tracking process.

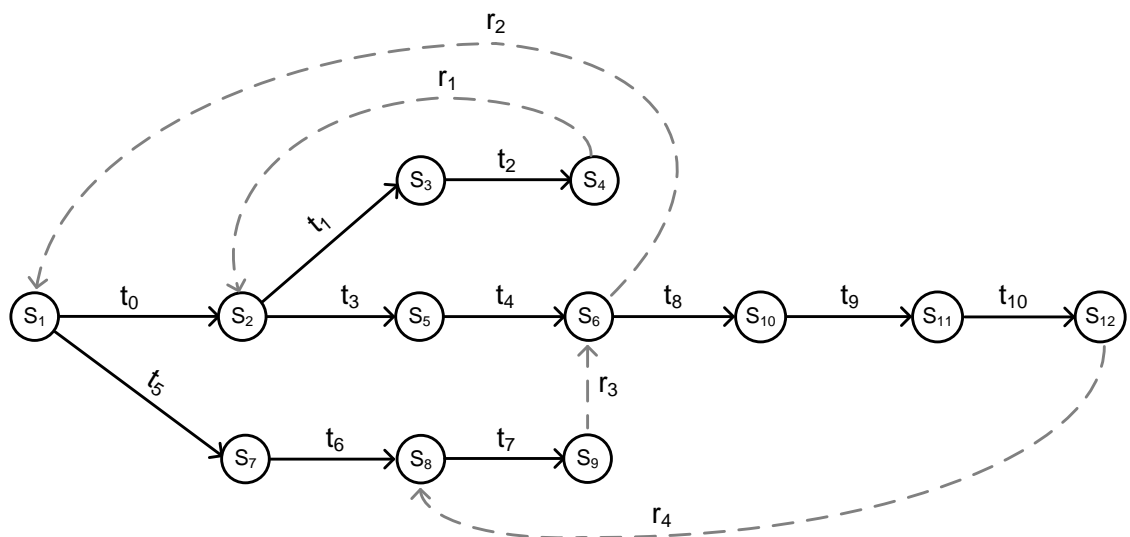


Figure 7.2: Example of Revert Operations

Figure 7.2 depicts a part of a creativity map with four revert operations. The revert sequences are marked through the dashed arrows. The number of the label indicates the order of appearance, starting with the lowest number.

The revert sequences of this creativity map are:

RS_1: \longrightarrow_{t_1} ; \longrightarrow_{t_1}

RS_2: \longrightarrow_{t_0} ; \longrightarrow_{t_3} ; \longrightarrow_{t_4}

RS_3: \longrightarrow_{t_5} ; \longrightarrow_{t_6} ; \longrightarrow_{t_7}

RS_4: \longrightarrow_{t_7} ; \longrightarrow_{t_8} ; \longrightarrow_{t_9} ; $\longrightarrow_{t_{10}}$

The identification of the transitions, leading to the first revert is straight forward. The sequence RS_1 consists of the transitions between the states S_2 and S_4 .

Sequence RS_2 is more complicated and seems to be a little odd. The transitions t_1 and t_2 are not elements of the sequence. The reason for this is the first revert RS_1. A rule is, that the, for a revert operation responsible, sequence will not contain branches. A branch always represents a former, unsuccessful approach.

This is the same case for sequence RS_3. The sequence contains only the transitions t_5 , t_6 and t_7 , because there was a revert from S_6 to S_1 before. The transitions depicted on the right, inside the figure, were created after the third revert r_3 . Those transitions therefore cannot belong to the sequence RS_3, as they were not existing as the time of the revert operation.

The last sequence RS_4 is a good example for the case that a revert sequence can contain transitions of different branches inside the creativity map. The first transition t_7 belongs to a different branch of the creativity map. This causes problems for the automated identification of the sequences with a computer.

7.3.1 Identification Technique

The first choice for identifying the sequences in a tree structure like the creativity map would usually be the Depth-first search (DFS). However, this approach cannot cope properly with the above mentioned problems. Furthermore, it would require traversing the whole creativity map, in case of a new revert operation. This is very inefficient and time-consuming in larger creativity maps. Therefore, it is not the best choice for the

identification of related transitions which are distributed over two different branches.

A much simpler and more effective approach is a table, which stores the start and end state of the revert sequence. Those two states are of course the old and new state of the revert operation.

	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10	S11	S12
RS_1		x		x								
RS_2	x					x						
RS_3						x			x			
RS_4								x				x

Table 7.1: Revert Tracking Table

Table 7.1 contains the information about the revert sequences of the example in Figure 7.2. The first row contains the identifier of the states, in this case $S1$ to $S12$. The following rows are added one by one with the appearance of a new revert. The two x inside each row stand for the start and end state of the sequence. The left “x” represents the start state of the sequence and the right one the end state.

The benefit of this table is the simplicity of spotting the transitions and states of the sequence, which are responsible for a revert operation. The resulting algorithm for the identification process is simple and fast as well. The identification of the first sequence is very easy. The sequence starts at $S2$ and ends at $S4$. For the following reverts, the algorithm will take the sequences from all previous revert operations into account. A previous revert sequence will be subtracted from the current one, if it is a subsequence. Through this, previous reverts are taken into account.

A former revert sequence can be identified as a subsequence with the tracking table, if the left and right “x” are between or in the same columns as of the current sequence. This is already the case for the second revert sequence, in the example table. One can see that the two “x” of the sequence $S2$ to $S4$ lie inside the sequence $S1$ to $S6$. Therefore, that sequence is not a part of the current revert operation. The states and transitions of the previous sequence are removed from the current sequence.

Now, the two subsequences of the current revert sequence need to be “glued” together at the right position. For this, the start state of the subsequence is connected to the state

on the right of the end state. In the example, $S2$ would be connected with $S5$, through the transition which created $S5$.

Nonetheless, this approach has one disadvantage when used for computer processing. It requires a lot of memory due to the size of the table. The most space is wasted, because the table is sparsely populated. Fortunately the approach can be adjusted in order to reduce the size of the table.

With a closer look at the table, it becomes obvious that the “x” marks a time of appearance. Furthermore, every state contains a timestamp of its creation. Hence, it is possible to create a table with only the timestamps of the start and end transitions.

	Start	End
RS_1	S2	S4
RS_2	S1	S6
RS_3	S6	S9
RS_4	S8	S12

Table 7.2: Improved Revert Tracking Table

The improved revert tracking table is much smaller than the original one. The first column contains the first or start state of the revert sequence. The second column contains the last or end state of the sequence. The number of columns is not related to the length of the revert sequences any more.

The identification process will compare the timestamps of the start and end state of the current revert sequence with the timestamps of the start and end state from the table. An old revert lies inside the current revert sequence, if the timestamp of the old start state is the same or more recent than the timestamp of the current start state and the timestamp of the old end state is older (not the same!) than the timestamp of the current end state. Already existing revert entries will not require a comparison again with new revert operations, as the new revert will contain states, which did not exist at the time the old one was created. A new revert sequence cannot be inside an old revert sequence. Only vice versa!

7.4 BPDFL Parser

The parsing of the creativity map for behavioural patterns is one of the most important steps of the analysis process. All information about the creative process is extracted through the parsing process. However, the parsing process is not limited to the creativity maps, but also caters for the analysis of the frequent behaviours. The patterns themselves are expressed through the Behavioural Pattern Description Language (BPDFL). The parser will read the expression and retrieve the adequate information.

The parsing and retrieval steps are quite complex and require a lot of preparation before any result is returned. The parser needs to create a logic representation of the possible sequences, a specific expression could provide as a result. A static approach is not useful as the outcome of the search is unknown. The reason for this is the flexibility of the BPDFL. Every creator can define own behavioural patterns and the resulting expressions. Hence, it is not possible to generate a simple list of expressions in advance. An approach which can cope with the creation of a logical representation of any BPDFL expression is required instead.

In computer science, this task is usually solved by using an automaton, which accepts a given expression in a language. The following pages of this chapter will describe techniques which enable the creation of an automaton for every possible expression of the BPDFL. The creation of the automaton requires several steps for the transformation of the expression into a DFA automaton. The transitions of the creativity map or frequent behaviours are the input for an automaton. Through this, the accepted sequences of an automaton will be returned as the result set of a behavioural pattern.

The application of a behavioural pattern on a creativity map is divided into five mandatory steps and one optional step. The six steps are in chronological order:

1. Preparation of the BPDFL expression.
2. Creation of a logical representation for the expression.
3. Creation of a NFA with ϵ -transitions.
4. Transformation of the NFA into a DFA.
5. Using the transitions of the creativity map as the input for the DFA.
6. Application of *SUM()*, *LEN()*, \sim , $-$ or *REVERT()* operations.

The first step is the analysis of the behavioural pattern expression. The second step handles the creation of a logical representation of the BPDFL expression. The step is required in order to map the order of the elements of an expression. This step is followed by the creation of a nondeterministic finite automaton (NFA), which accepts all possible sequences, described by the behavioural pattern. The NFA allows ϵ -transitions, which simplifies the creation of the automaton. The fourth step is the transformation of the NFA into a deterministic finite automaton (DFA). This is necessary as the transitions from one state to another are not explicit in a NFA. The DFA is used in the fifth step for the extraction of sequences, matching to the behavioural pattern. The sixth step is optional and only required in case the expression contains a *SUM()*, *LEN()*, \sim , $-$ or *REVERT()* operation.

7.4.1 Structures and Preparation

The information extraction process via the BPDFL requires a common structure of the creativity map for an easy access of the states and transitions. The data structure of the creativity map has much influence on the processing speed of the analyses. A powerful, but simple structure is required for an efficient storage of the data and information as well as simple access to the relation of the states and transitions.

Furthermore, it is necessary to prepare the BPDFL expression, to make the required information easily accessible. This will reduce the complexity of the transformation process for the expression.

Adjacency List

The transitions of the creativity map need to be stored in a special structure. A graph or tree structure is usually stored in an adjacency list or adjacency matrix. Creativity maps are stored in an adjacency list, as this structure is sufficient for the store of a tree structure. Furthermore, it saves resources in regard to an adjacency matrix. The size of an adjacency matrix depends on the number of states in a graph, regardless how many transitions it contains. This is rather inefficient for a tree structure, like the creativity map, which furthermore contains usually long sequences. The adjacency matrix would be sparsely populated in this case. In contrast to this, an adjacency list always contains only required information.

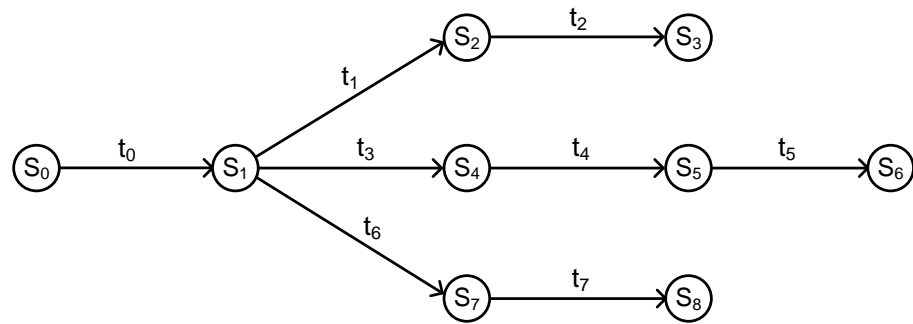


Figure 7.3: Creativity Map Example

Figure 7.3 depicts a small example of a creativity map. It contains nine states ($S_0 - S_8$). The owner of this creativity map made two reverts during his creative process. Hence, there are three branches inside the creativity map.

The adjacency list of this creativity map must describe all relations of the states.

State	Successors
S_0	S_1
S_1	S_2, S_4, S_7
S_2	S_3
S_4	S_5
S_5	S_6
S_7	S_8

Table 7.3: Adjacency list of a creativity map

Table 7.3 illustrates the simplicity of the adjacency list. The table always consists of two columns. The first column describes a state with leaving transitions. The second column contains the states, which are the direct successors of the state in the first column.

The table above enables the creation of the creativity map depicted in Figure 7.3. Only states, which have a successor are listed in the first row. Nonetheless, every state is present in the table as every state must be the successor of another state, if it is not the initial state.

NB, the adjacency list contains no information about the revert sequences. This explains the need for the separate storage of information about the revert processes.

The adjacency list needs to be modified for the pattern identification framework. A normal adjacency list contains no information about the transition or edge which leads from one state to another. Hence it is impossible to extract the details about transition types and viewpoints.

The modified adjacency list is realised through a state object, which contains a list of the leaving transitions of that state. The transition objects in the list contain information about the start and end state, label and viewpoint of a transition. This approach is very similar to the standard adjacency list, but enriched with additional information, which is required in the parsing and analysis process.

Preparation of the BPDFL expression

The preparation of the expression is quite simple. First it is necessary to remove the blanks (spaces) inside the expression. This simplifies the following recognition of the expression.

$\langle pattern \rangle := \sqcup \langle c_transition \rangle \sqcup \langle a_transition \rangle ;$

The example shows a BPDFL expression with spaces in between. Here represented by the \sqcup symbol. Another creator might not put in any spaces at all. Sometimes a creator might forget to set a space between two elements of the expression, while having used some before. The parser must be able to process the expression correctly in all cases. The removal of the spaces is keeping the processing routines simple.

In the second step, following parts will be removed from the expression:

“ $\langle pattern \rangle :=$ ” and “ $;$ ”

The core of the expression,

$\langle c_transition \rangle \langle a_transition \rangle$

is then used for the further parsing process. The step is necessary as the removed elements contain no valuable information for the parsing process.

7.4.2 Logical Representation of BPDFL Expressions

The creation of an automaton for accepting a BPDFL expression requires the transformation of the expression into a logical structure. Without such a structure, it would be impossible to map the structure and arrangement of the elements of the expression. The meaning of the expression must remain the same through all of the following transformation steps.

The structure must enable the smooth creation of an automaton for the given expression. The most common way of representing a regular expression or similar structures is realised through a tree structure [60].

The structure of the so-called parse tree is directly transformed into a NFA. In such a tree, the transition types are represented by the leaves of the tree. The constructs like repetitions, choices and sequences are defined by the nodes of the tree. The root element of a tree is always a sequence node.

Following example of a BPDFL expression,

$$\langle c_transition \rangle \langle a_transition \rangle | \{ \langle c_transition \rangle \}$$

is transformed into the following parse tree.

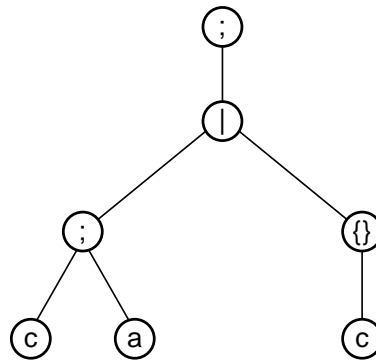


Figure 7.4: Parse Tree

Figure 7.4 depicts the parse tree for the BPDFL expression. Every element of the expression can be found inside the parse tree. The structure of the expression is added to the mandatory root sequence element.

Every element of the expression is recognised by its first character. This is possible because every element and operator starts with a different character. The necessary remaining characters of the element or structure (brackets or curly brackets) are discovered afterwards. The process will then read the character following after the last character of the element for the identification of the next element. This is repeated till the end of the expression. The elements of an expression are read step by step.

The algorithm for the creation of a parse tree recognises the different elements of the expression and adds an equivalent node to the parse tree. The process will handle the different element of the language as follows.

- transition: Add a leave to the parent node, with the label of the transition.
- repetition: Add a repetition node to the parent transition, parse the content of the curly brackets recursively and append the resulting sub-tree below the repetition node.
- choice: Create a choice node and move the children from the parent node to this node. Add the choice node as a child of the parent node. Set the choice node as the parent for the following nodes.
- normal brackets: Add a sequence node to the parent node and recursively parse the contents of the brackets. Add the resulting nodes as children to the sequence node.

The list is of course not complete and a simplification of the whole reading process. The special operators like *SUM()*, *LEN()*, *REVERT()*, without ($\backslash(x, \dots)$), branching (\sim) and merging ($-$) are missing at this stage. It was already mentioned that most of those functions are not handled by the automaton itself, but the applied on the results of the automaton processing.

The information about the without ($\backslash(x, \dots)$), branching (\sim) and merging ($-$) operations are stored inside the nodes of the parse tree. The information will be kept inside each automaton during the next transformation steps. This information is required during the analysis process, when traversing the creativity map.

The creation of the parse tree is finished with the identification of the last element of the expression. The tree requires no further preparation and is ready to be used for the creation of a NFA.

7.4.3 Thompson's Construction

The nondeterministic finite automaton (NFA) is created through a bottom-up approach. The so-called Thompson's Construction is used to build a NFA through a set of basic structures [124]. These basic structures are combined through ϵ -transitions in order to create a more complex automaton, which will accept all possible sequences of a behavioural pattern. The here presented approach is a well-known standard procedure.

The Thompson's Construction is used to generate a NFA from a regular expression or in this case a BPDFL expression. This allows the creation of an automaton which accepts the possible sequences described by an BPDFL expression. The following figure contains the four basic structure of the language. These are the normal transition, the sequence, the repetition and the choice (or).

The elements are created through a recursive transformation of the parse tree. This technique enables the creation of the NFA through the before mentioned bottom-up approach.

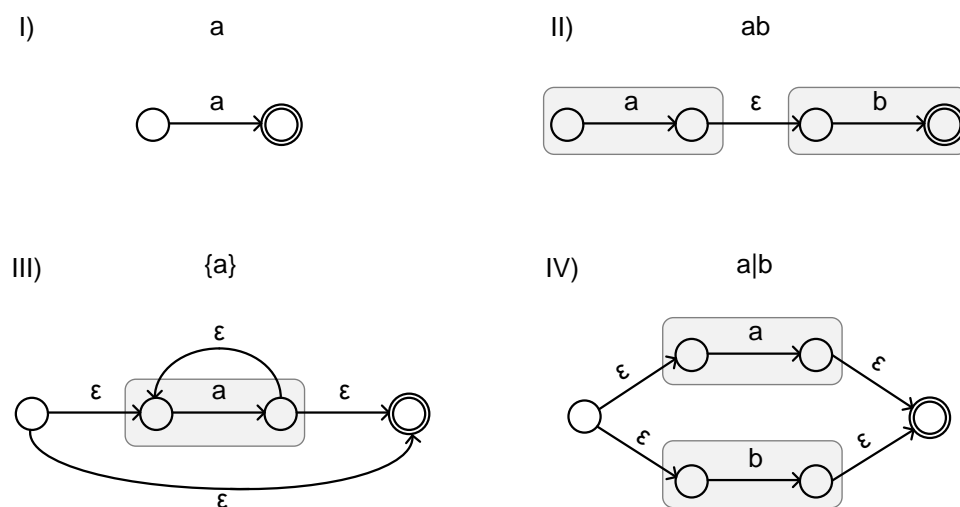


Figure 7.5: Basic Constructs for the NFA

Figure 7.5 depicts the common constructs of the NFA [28]. It is important to note that the depicted constructs are not pieces of a creativity map. The transitions of a creativity map will be the later input for the NFA.

It is possible to represent all valid variants of BPDFL expressions through a combination of these common constructs, apart from the special operators, like the REVERT() operator. The first structure is the simplest one. It represents the acceptance of a word, in this example “a”. This structure reoccurs in all other structures. It is the basic construct in all automata and is marked by a grey box within the other structures.

The second structure (II) represents a sequence of two transitions a and b . A sequence is represented inside the NFA through connecting the two basic transition constructs through an ϵ -transition.

The creation of the third construct (III), a repetition, is more complex. The construct requires two new states, one as the initial and one as the final state of the construct. The basic construct lies in between and is connected to the initial and final state through ϵ -transitions. One ϵ -transition enables to represent a repetition of zero through connecting the initial and final state. Another ϵ -transition enables two or more repetitions by moving back from the end to the start of the basic structure.

The fourth and last construct (IV) enables the representation of a choice between two elements. The construct contains two basic structures, one accepts the word “a” and the other accepts the word “b”. Furthermore two additional states, where one becomes the new initial and one the new final state of the choice construct. The new states and the two basic structures are connected through ϵ -transitions.

The presented structures can be obviously used for the creation of an automaton which accepts an expression, describing a behavioural pattern. The following example illustrates how the basic structures are used for the creation of an NFA, which accepts a more complex sequence.

Example for Thompson’s Construction

The example from before will be used again: It has following expression:

$$\langle c_transition \rangle \langle a_transition \rangle \mid \{ \langle c_transition \rangle \}$$

The parse tree for this expression is depicted in Figure 7.4. The creation of the equivalent NFA for this expression will be created in three steps. The number of steps depends on the nesting of operators and sequences. The creation process starts with the smallest constructs, which will be combined, regarding the presence of operators.

Creation of the Basic Structures

The basic structures are always created with the transition types inside the BPDFL expression. This can be more than two types in case there are different chosen-transitions in the expression. A special type is of course the any-transition which accepts every word (transition in the creativity map).

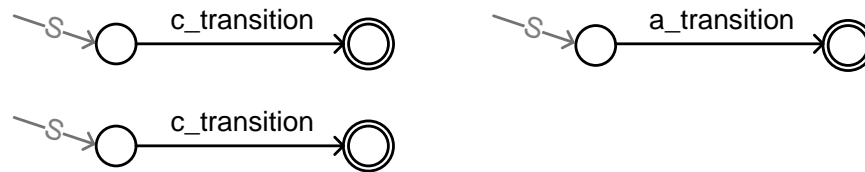


Figure 7.6: Basic Constructs of the NFA Example

The expression contains three transitions. Figure 7.6 depicts the representation of these transitions in the NFA. The grey arrows indicate the initial state. The final state is indicated by a double ring. These constructs are used for the next step of creating the NFA.

Creation of the Sequence and Repetition Structures

This step uses the constructs with the next applicable operations. The first operation is the creation of a construct which represents the sequence $\langle a_transition \rangle \langle c_transition \rangle$.

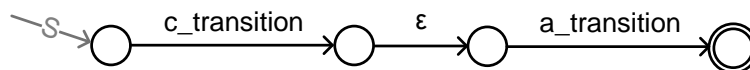


Figure 7.7: Sequence Construct of the NFA Example

Two of the three basic constructs have been connected through an ϵ -transition. Only one state remains as the initial state after combining two basic constructs in a sequence construct. The same happens with the final states. One can already imagine the simplicity of this approach, which makes it easy to transform it into an algorithm.

The last basic construct is used for the creation of a repetition construct.

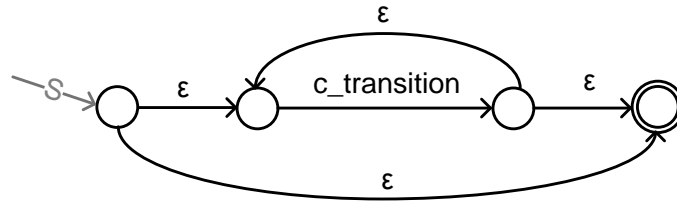


Figure 7.8: Repetition Construct of the NFA Example

The second $\langle c_transition \rangle$ is enclosed inside a repetition construct. The construct is seamlessly integrated into the construct via the ϵ -transitions. The initial and final state are changed to the outer states of the repetition construct.

Creation of the Choice Structure

The last operation left inside the expression is the choice “|” operation. The NFA will be complete after the creation of this construct.

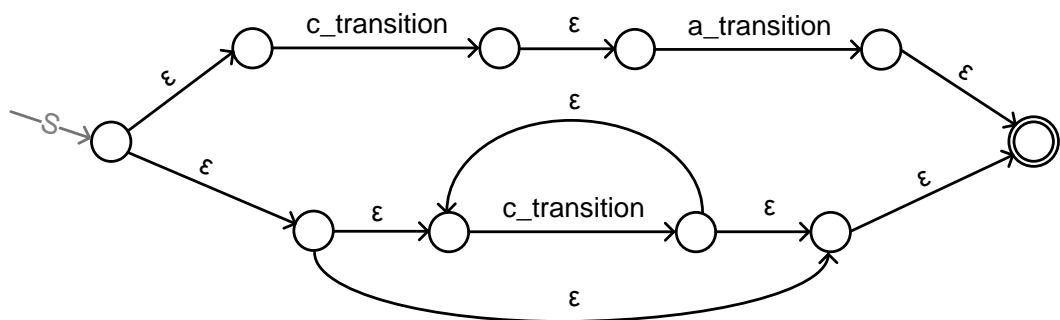


Figure 7.9: Choice Construct of the NFA Example

The choice construct combines the repetition and sequence construct from above. The result is a NFA which accepts the BPDFL expression of the example. In a next step, it is necessary to transform this NFA into a DFA, in order to get rid of the ϵ -transitions.

7.4.4 Subset Construction

It was already mentioned that only a DFA has explicit transitions from one state to another. Therefore, it is not possible to use the NFA with ϵ -transitions for the task of accepting a BPDFL expression. However, it is possible to transform a NFA into a DFA. As a general rule: There exists always an equivalent DFA for every NFA.

A direct creation of a DFA without creating a NFA first is not favourable as the NFA with ϵ -transitions is much easier to create. Especially when combining sub-constructs into one automaton.

The subset construction, also known as the powerset construction, is used for the transformation of a NFA into a DFA. The transformation process is based on a transition table, which uses sets of states of the NFA as the input. These sets are identified through the usage of two functions. These are the ϵ -closure() and move() functions.

The two functions are defined as following:

- ϵ -closure(S_i), creates a set, containing all states which are reachable from S_i through an ϵ -transition.
- move(S_i, t), creates a set, containing all states which are reachable from S_i through a specified transition type t .

The function ϵ -closure(S_i) is necessary for the elimination of the ϵ -transitions in the NFA. This is mandatory for the identification of all states, which can be reached by a certain word or transition type. The function move() is always followed by the ϵ -closure in order to reach all states belonging to the set.

The subset construction defines some basic properties of the DFA in regard to the relation of the NFA, describing the same language. These properties are:

- The DFA and NFA share the same alphabet.
- The states of the DFA are sets of states from the NFA.
- The initial state of the DFA is the initial state of the NFA plus its ϵ -closure.
- The final states of the DFA are those, where a set contains a final state of the NFA.

State	NFA-states	Transition Type	
		c	a
S_0			
S_1			
S_2			
\vdots			
S_n			

Table 7.4: Transformation Table

Table 7.4 depicts the transformation table for the subset construction. The table consists of three major parts; the new state of the DFA; the NFA-states and the columns with the transition types. The entries of NFA-states contain the states which are later represented by the new DFA state. The number of rows is not dependent on the number of different transition types, but the number of states and the complexity of the NFA.

The process of filling the table is as follows:

1. Identify the initial state of the NFA and apply the ϵ -closure of this state. The resulting set of states is used for the first state of the DFA. The first is also the initial state of the new DFA.
2. Apply the $move()$ function on the above mentioned set of states. The results are stored in a set and used as the input for the ϵ -closure. The step is executed for every transition type and the results of the $move()$ and ϵ -closure are stored in the column of the transition type.
3. Check whether the newly identified sets appear at another position of the NFA-states column or is empty. For this, start with the first transition column of the current row. In case it is not empty and did not appear in the NFA-states column before, then a new row is added to the transformation table and the set is copied to the cell of the NFA-states column. This process is repeated with all following columns of the current row.
4. Jump to the next (new created) row and start again with step two, using the NFA-states of that row. The process is complete when no new NFA-states are found and then end of the transformation table is reached.

The DFA can be created as soon as the last row of the transformation table is completed. The last step is the identification of final states. It was already mentioned that a state of the DFA is final, if the set in the column NFA-states contains a final state of the NFA.

Example for the Subset Construction

The result of the example for explaining the Thompson's Construction is used for the explanation of the subset construction.

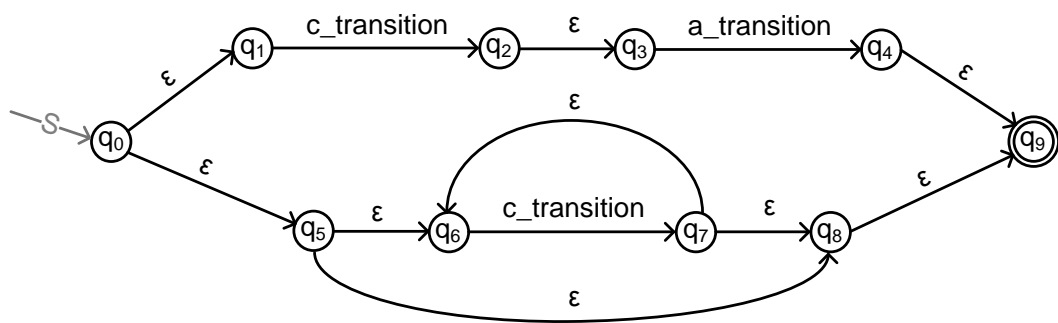


Figure 7.10: Input for the Subset Construction

Figure 7.10 depicts the NFA of the earlier example, which is now transformed into a DFA. The labels of the states have been changed in order to illustrate that the states of the NFA and the resulting DFA are different.

State	NFA-states	Transition Type	
		<i>c_transition</i>	<i>a_transition</i>
S_0	$q_0, q_1, q_5, q_6, q_8, \underline{q_9}$	$q_2, q_3, q_7, q_6, q_8, q_9$	-
S_1	$q_2, q_3, q_7, q_6, q_8, \underline{q_9}$	q_7, q_6, q_8, q_9	q_4, q_9
S_2	$q_7, q_6, q_8, \underline{q_9}$	q_7, q_6, q_8, q_9	-
S_3	$q_4, \underline{q_9}$	-	-

Table 7.5: Transformation Table Example

The in Table 7.5 depicted transformation table contains the details for the creation of the DFA. The initial state of the example NFA is q_0 . Nonetheless, its ϵ -transitions allow reaching the states q_1, q_5, q_6, q_8, q_9 . The last state of the set is underlined, which indicates a final state. As a result, state S_0 will be a final state as well.

Now, the starting set is used for step two of the transformation process. The column *c_transition* contains all states which can be reached via the $move(q_n, "c_transition")$ function, followed by the ϵ -closure(q_n) function. The result set contains six states: $q_2, q_3, q_7, q_6, q_8, q_9$. The same step for the transition type *a_transition* will result in an empty set as no state can be reached through this transition type.

After finishing the the steps for the whole row, the result sets are used to create new rows in the transformation table. In this case only the first set.

The row of state S_2 is very interesting. The set of the cell for the transition type *c_transition* is identical to the cell of the column NFA-states. Hence the state will have a *c_transition* pointing to itself.

The transformation process finished with the row of state S_3 after no new sets of transitions could be found.

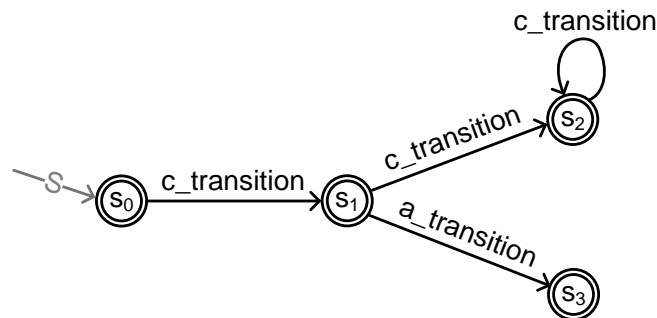


Figure 7.11: Result of the Subset Construction

The creation of the DFA is based on the appearance of the NFA state sets in the transformation table. Figure 7.11 depicts the result of the subset construction. In this example, every state of this automaton is a final state. The transition above state S_2 represents a loop, enabling repetitions of the word *c_transition*.

This automaton is now ready to be used for the identification of the behavioural pattern. It will accept all valid sequences for this pattern.

7.4.5 Usage of the Automaton

The creation of the DFA marks the final stage of the transformation of the BPDFL expression into a logical representation. Unlike the NFA, the DFA is directly used for the analysis of the creative behaviour.

The identification of a behavioural pattern, described through a BPDFL expression, is based on the transitions of the creativity map as the input for the automaton. The algorithm for the pattern identification process will start with the first transition of the creativity map and check if it is accepted by the automaton. The algorithm will then continue with the process, depending whether the first element was accepted or not.

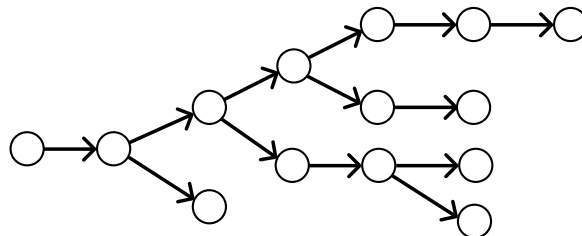


Figure 7.12: Creativity Map Example

Figure 7.12 depicts a small example for a creativity map. The algorithm will traverse the tree with the Depth-first search (DFS), starting with the first transition of the creativity map. The selected transition is then used for the identification of a behavioural pattern. For this, a second DFS process will be initiated to traverse the sub-graph, starting with the before mentioned transition.

The transitions, which are selected by the second DFS, are used as the input for the automaton of the behavioural pattern. An entity of a behavioural pattern is found, if a sequence is accepted by the automaton.

The algorithm for the acceptance of sequence, matching to a behavioural pattern is illustrated in the following two flowcharts. The described algorithm is providing the above mentioned capabilities. Advanced operations, like the branching operator are not included. These will be described separately for a better understanding.

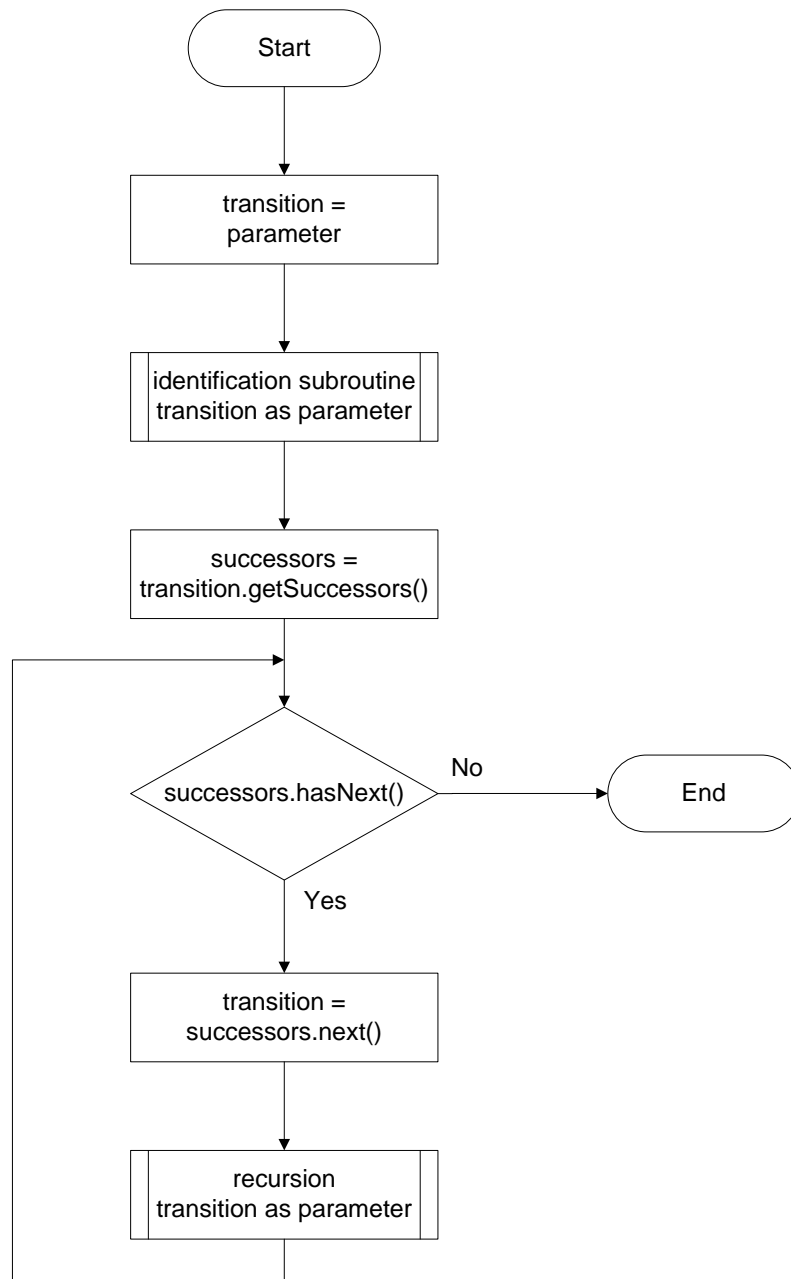


Figure 7.13: Algorithm for the Traversing of the Creativity Map

The flowchart in Figure 7.13 depicts the DFS for the traversing of the creativity map. The depicted algorithm is used to traverse the complete creativity map once. It is used to pick up every transition of the creativity map for the behavioural pattern identification process, which is depicted in the next flowchart.

The value of the parameter, in the second block of the flowchart, is first transition of the creativity map. It is needed to start the process from the root of the creativity map.

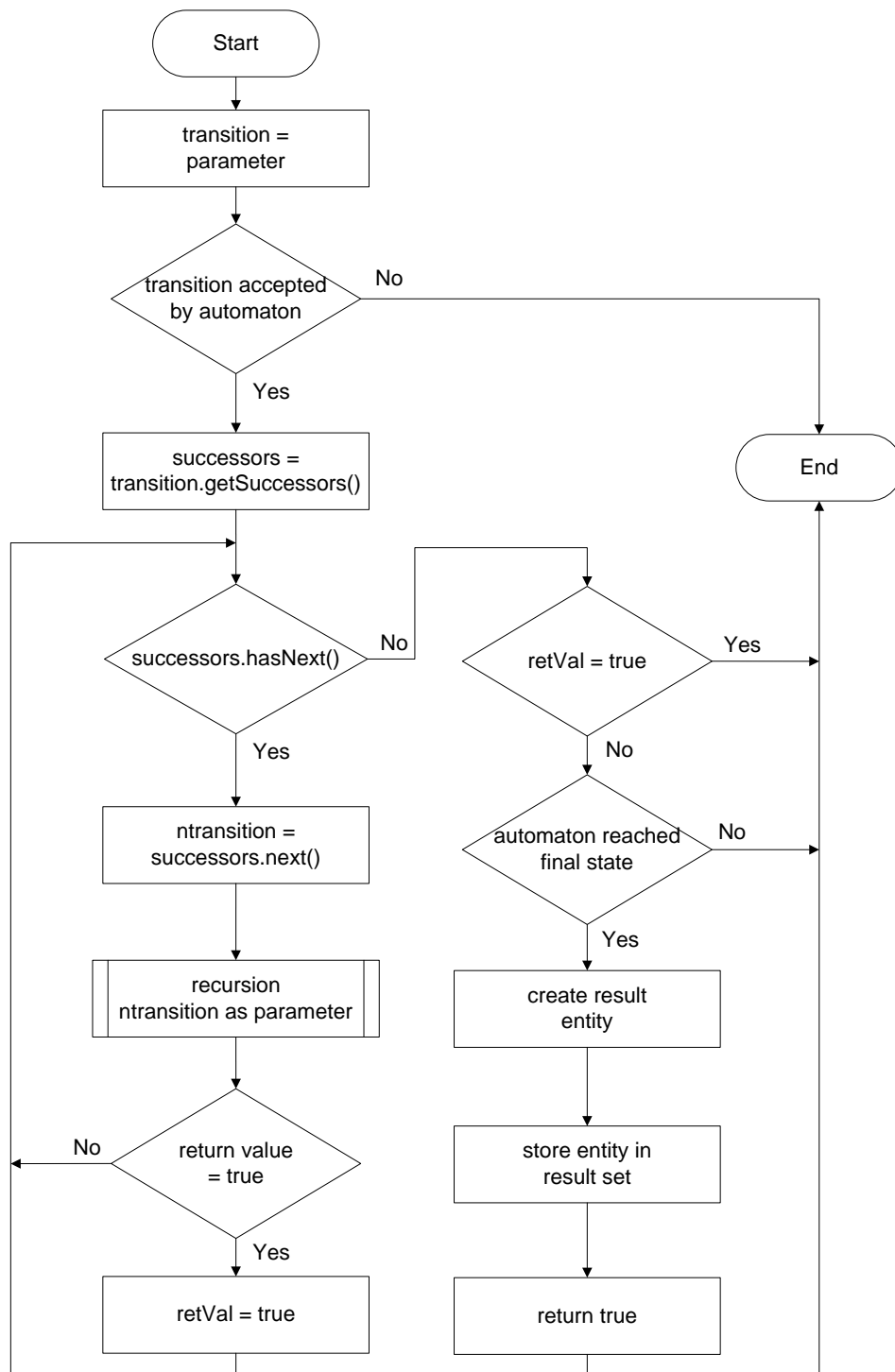


Figure 7.14: Sub-routine for the Identification of Behavioural Patterns

Figure 7.14 depicts the steps for the identification of sequences, matching a behavioural pattern inside the creativity map. The depicted algorithm uses the transition, which is selected by the traversing algorithm. Both algorithms use recursions for the traversing through the creativity map. The result of the identification process can undergo further processing, in case the BPDFL contains further instructions.

The variable *transition* inside the flowchart represents the currently used transition, which was selected through the first DFS or the parent instance, in case of a recursion. The second variable *successors* contains the transitions, which are direct successors of the current transition. The transitions and states of each result entity are stored in a vector. This vector is added to another vector, which contains all results of the parsing process for the given behavioural pattern. The two flowcharts do not cover every aspect of the algorithms, but certainly the important ones.

The algorithm will always select the longest possible result for a behavioural pattern. This benefits the analysis of the creative behaviour, because longer sequences usually contain more information about the creative process than small ones.

It is not always necessary to start with the first element of the creativity graph and to proceed with the consecutive transitions. The algorithm can directly search for a certain transition type, in case the first element of a BPDFL expression is a chosen-transition. This can reduce the number of attempts drastically. Of course, this is only possible if the chosen-transition is definitely the first element of the expression.

7.4.6 Advanced Operations

There is a number of special operators within the BPDFL, namely the *LEN()*, *SUM()*, *REVERT()*, branching and merging operators, which are not covered by the automaton. The complexity of those operations could not be handled by an automaton or only with an enormous overhead. Instead, these operators are applied on the results of the automaton step.

It was mentioned, that every result entity of the parsing process is stored inside a vector. This data structure is very beneficial for the further processing of the results. The data structure not only enables the storage of sequence. It is furthermore possible to store tree structures, which can emerge from the usage of branching and merging operators.

The following paragraphs provide details about the design of the advanced operations.

LEN()() Operator

The $LEN()()$ operator uses a simple approach of calculating the length of a result. The algorithm for the operator counts the number of transitions of a result entity. This process is directly undertaken after the identification of the sequence. Only sequences, which satisfy the condition, are added to the result set.

The immediate check of the length after the creation of the sequence provides a benefit in case a behavioural pattern can return several results from a single path. Usually, the longest possible sequence will be returned as the result. This function can be influenced by the $LEN()()$ operator, in order to get a different result. For example if the pattern expression contains a repetition of one or more transitions.

The $LEN()()$ operator enables to simplify the following expression:

$$\begin{aligned} pattern := (< a_transition > < a_transition > < a_transition >) \\ & | (< a_transition > < a_transition >) \\ & | (< a_transition >); \end{aligned}$$

into

$$pattern := LEN(\{< a_transition >\})(< 4);$$

In this expression, the repetition operation would return the longest possible sequence of a path. The $LEN()()$ operator however will only allow results with less than four transitions. The parser will, therefore, return a sequence with the length of three transitions, in case the longest possible sequence contained more than three transitions.

It was mentioned before that this operator can only be used at one position of the BPDFL expression. The operator is only to be applied on the whole expression.

Branching (\sim) and Merging ($-$) Operators

The automaton for any BPDFL expression is not capable of handling branches. Therefore, the automaton requires a function to mark the position of each occurrence of the two operators inside the sequences of the result set. Through this, it is later possible to identify the states and transitions which need to be “glued” together or kept in parallel in order to create branches.

The two operators will trigger different actions, despite giving the impression of being very similar. The merging operation itself will not produce any different results than it would without the operator. The effect of the merging is only visible when applied together with the SUM() operator. The reason for this was already discussed in Chapter 4: A state will not have multiple incoming transitions. As a result, the algorithm for the merging operation will only mark the position of the merging operator at the entities of the result set. The marks are used by the SUM() operator for the proper creation of the single result entity.

The behaviour of the branching operator is different, because a state can have multiple leaving transitions. Hence, the occurrence of branches is possible without the usage of the SUM() operator. The creation of the branches is immediately performed after the identification of all matches of a behavioural pattern for a specific start transition, selected by the first DFS routine. The process selects the first entity and appends the required states and transitions from the other result entities. After this step, only the first entity is returned as a result.

SUM() Operator

The SUM() operator is the last one to apply on the result set. It will only operate with the complete result set of the parsing process. The algorithm for this operator will search the result set for entities, which can be combined. The operator cannot combine any results without the usage of either the merging or branching operator.

It was mentioned before, that the position of the branching and merging operator is marked in every entity of the result set. This information is used by the SUM() operator to create the combined results.

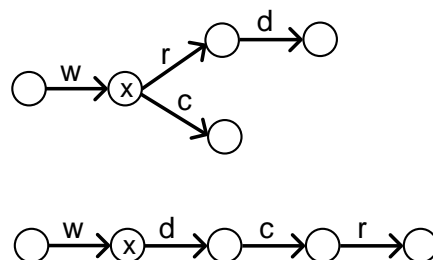


Figure 7.15: Two Entities with Branches

Figure 7.15 depicts two entities of a behavioural pattern with a branching operator. The position of the branching operator is marked by a “x”. The algorithm for the SUM() operator will check the entities for this marking.

The algorithm will check if the sequences before the marked states of the two entities are the same, in case of a branching operator. If this is given, then the parts of the entities behind the marked states will be combined and appended at a new entity, consisting of the sequence before the marked states. In case of the branching operator, the algorithm will do the opposite. It will check the transitions behind the marked states and combine the parts before the marked state. The original entities will be deleted.

Same as the *LEN()* operator, this operator can only be used at one position of the BPDFL expression. This operator is also only applied on the whole expression. It is possible to use the branching and merging operators at the same time.

REVERT() Operator

The *REVERT()* operation can be used in four different ways. A chosen-transition as parameter is allowed in all cases. However, it is not allowed to use the operator within a nesting, like repetitions or brackets. The following cases are valid:

- I *pattern := sequence REVERT();*
- II *pattern := REVERT() sequence;*
- III *pattern := sequence REVERT() sequence;*
- IV *pattern := REVERT();*

The algorithm for this operator will not require any marking of the automaton or similar actions. It creates an automaton for the sequence or sequences and will compare the results from using the automaton(s) with the results of the *REVERT()* operator.

The comparison is based on the timestamps of the states. For example, for case no. I: A result can be returned, if the timestamp of the end state of the last transition for the sequence is same as the timestamp of the start state of the first transition, given by the *REVERT()* operator. Case no. IV will return all reverts for a creativity map. No further preparation is required.

This process is using the information from the revert tracking process. Therefore, an access to the database, storing that information, is required.

7.5 Monitoring Process

The monitoring process for the identification of a certain behaviour during the current creative process of a creator is almost similar to the analysis of the complete creativity map. The analysis uses the current input of an user, instead of the complete data from the creativity map. This means a transition will be checked by the parser at the moment of its creation.

It is possible to monitor for several behavioural patterns at the same time, as the parsing framework is supporting the creation and usage of several automatons at the same time. The monitoring process will return the identified sequence after the occurrence of a behavioural pattern. Furthermore, the creator will be informed about which behavioural pattern was identified, in case the process is searching for a number of patterns.

The algorithm for the monitoring process requires no processing of branches. The reason for this is the fact that only the current creative path is used as the input data. The longest possible path for the analysis would be from the initial transition of the creativity map to the latest transition.

The monitoring process is not completely covered by the BPDFL parser. The method only checks whether a given transition is accepted by the automaton. The monitoring process is using a step-by-step approach, where only one transition is used at a time. The method can return three different messages to the calling instance: *not accepted*, *accepted* and *pattern recognised*.

The method progresses in the automaton with every new accepted transition. The first call of the method will start at the initial state of the automaton and check, if the transition type is accepted. If this is the case, the method will then return the information that the transition was accepted and use the reached state of the automaton for the next call of the method. The process is repeated with the newly reached step until a final state of the automaton is reached or the transition is not accepted.

The monitoring process will always directly signal the identification of a behavioural pattern and does not wait if the automaton will accept a longer sequence. This is very important in case the behavioural pattern describes poor creative behaviour.

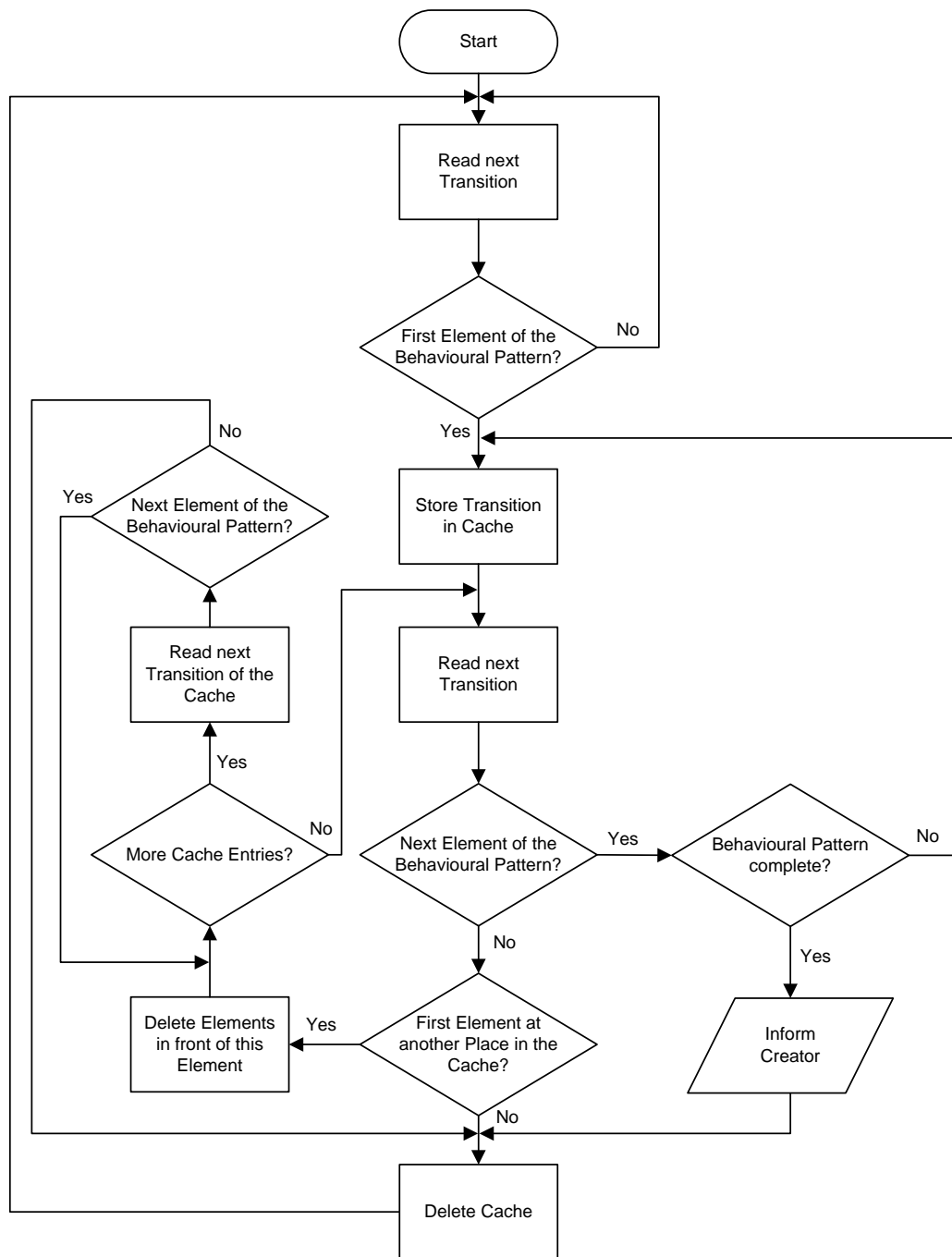


Figure 7.16: Monitoring of Current Behaviour

Figure 7.16 depicts the flow chart of the complete algorithm for the monitoring process. The flow chart has a starting point but not an end. An end element was not introduced in order to show that the monitoring process is not ending, but always waiting for the next activity of the creator and therefore a new transition for the creativity map.

The monitoring process needs to cater for two special events which will happen on a regular basis. These events make it necessary to re-read a certain number of transitions. The events or causes are triggered by a positive identification of a behavioural pattern or a revert operation.

It is important to note that the process will continue with the second transition of the identified entity of a behavioural pattern and not the next transition following after the entity. This is necessary, because every transition inside this entity could be the first transition of another entity of the same behavioural pattern.

The process needs to store the first transition of every identified entity of the behavioural pattern. This transition will be used during the initialisation of the framework itself. The monitoring process needs to re-read all the steps since the last identification to cover the case that the sequence starting from there contains the first few transitions of another entity. The same procedure is necessary for the revert operation. This operation is the reason, why the first transitions of all entities must be stored, because a revert operation can cause a “jump” to every previous state of the creativity map.

The implementation of the monitoring process is designed to not directly initiate any actions. The reason behind this approach is the idea that the usage of the results is kept open and not bound to a specific action.

7.6 De Montfort Creative Environment

The De Montfort Creative Environment (DMCE) is a set of tools, consisting of the client side applications. It represents the client side of the DMCA. The applications itself are loosely coupled to provide full flexibility for the user. The software is written in Java, which makes it platform-independent as long as Java is available and installed on the target system [22].

The current version of the environment contains applications, specially designed for creative writing projects. The tool set includes a project management facility, instant messaging with an integrated whiteboard, pend-it notes with a reminder functionality and a real-time collaborative text-based editor. The design of the environment follows the principles of normal desktop applications.

The following pages will provide details about the applications and their functionality.

7.6.1 Project Management Facility

The project management facility is the first window, which opens after the successful login to the system. It is the main application for the management of the projects and documents as well as to launch the other applications of the DMCE.

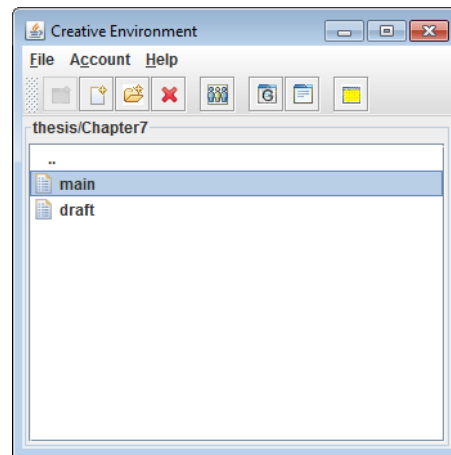


Figure 7.17: Project Management Facility

Figure 7.17 depicts the project management facility. The design of the tool is kept simple, in order to avoid a long adaptation phase.

The centre and bottom of the window contains a file browser. It enables to access the hierarchical file system. The structure of the file system knows three different elements: project, directory and file. The root element is always the project. A project can contain directories or files. The directories can contain other directories or files. The files itself are not real files, but links to a file, stored on the DMCA server. Users of the system can invite other users to join their project in order to collaborate. The projects of others, where the user has been invited to contribute, are also listed in the same window, with a slightly different appearance for a better distinction.

The project management facility contains a toolbar with a number of buttons, which launch other applications of the DMCE like the instant messaging facility, Pent-it Notes and the access control management tool. The collaborative editor is opened through a double-click on a document inside the project browser pane.

The project management facility is always opened during the sessions of the user. If closed, all remaining application will be closed automatically as well.

7.6.2 Collaborative Editor

The collaborative editor represents the main application, which is used to work on the creative artefact. In this case a text document. For other domains, this application might be replaced by another application. This could be e.g. a tool to create paintings or music.

The collaborative editor was designed to provide a simple, but powerful application for the creation of creative literature. It contains some special features to address the special needs of the users of such an application.

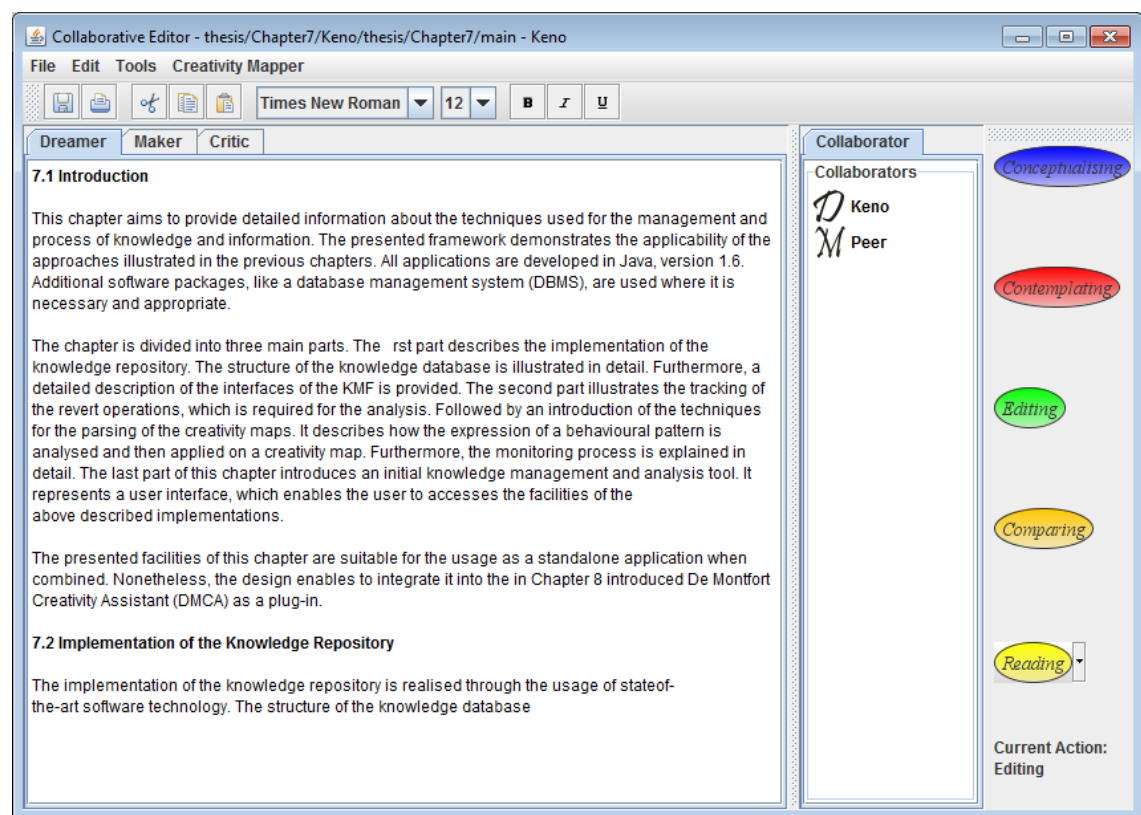


Figure 7.18: Collaborative Editor

Figure 7.18 depicts the collaborative editor. One can see that it looks similar to other well known editors like WordPad from Microsoft or Gedit for Linux. The user can choose different fonts and sizes, import pictures and much more. However, the collaborative editor has three additional features, which distinguishes it from other editors.

The first one represents a list of names, which contains the names of the collaborators, working on the same document. It is possible to show the contributions of each collaborator by “double-clicking“ on the name in the list. After this, the parts of the text, written by this person, will be highlighted.

Another additional feature is located on the right side of the name list. It is a tool bar with a number of different, oval-shaped buttons. The tool bar represents the input generator for the DMCM. Each button carries the name of an activity. The tool bar and the buttons itself are fully customisable. A more detailed introduction into this tool is given later in this chapter.

The last additional feature of the collaborative editor consists of three different writing areas. These can be selected via tabs above the writing area. The writing areas are required for a special method in creative writing - the so-called Dreamer/Maker/Critic modes.

The three labels describe different stage of the writing process as well as different states of the creator’s mind.

- Dreamer

The dreamer mode is mostly used during the beginning of the writing process. It represents the stage where the creator has not yet found a clear structure and is imagining and creating a storyboard.

- Maker

The maker mode is following after the dreamer mode. However, the writer will often jump between the modes. This mode describes the stage where the writer creates the content in a form, which is very close to the publishing stage.

- Critic

The final stage is the critic mode where the writer critically reviews the document. At this stage, there will be only minor changes to the document. The end of this mode will emerge the final document.

This approach is widely known and belongs to the repertoire of many writers and is therefore often taught in faculties with courses for creative writing and new media.

7.6.3 Text Analysis Facilities

The version control system of the DMCA enables a comparison of different stages of an artefact. In the case of creative writing, it enables the comparison of different versions of the document. Furthermore, it is not only possible to compare different versions, but to restore a previous version of the same artefact and continue working on it. The collaborative editor of the DMCE contains an integrated text analysis facility.

The text analysis facility of the collaborative editor is started through the menu entry *Tools* → *Show Revision History*.

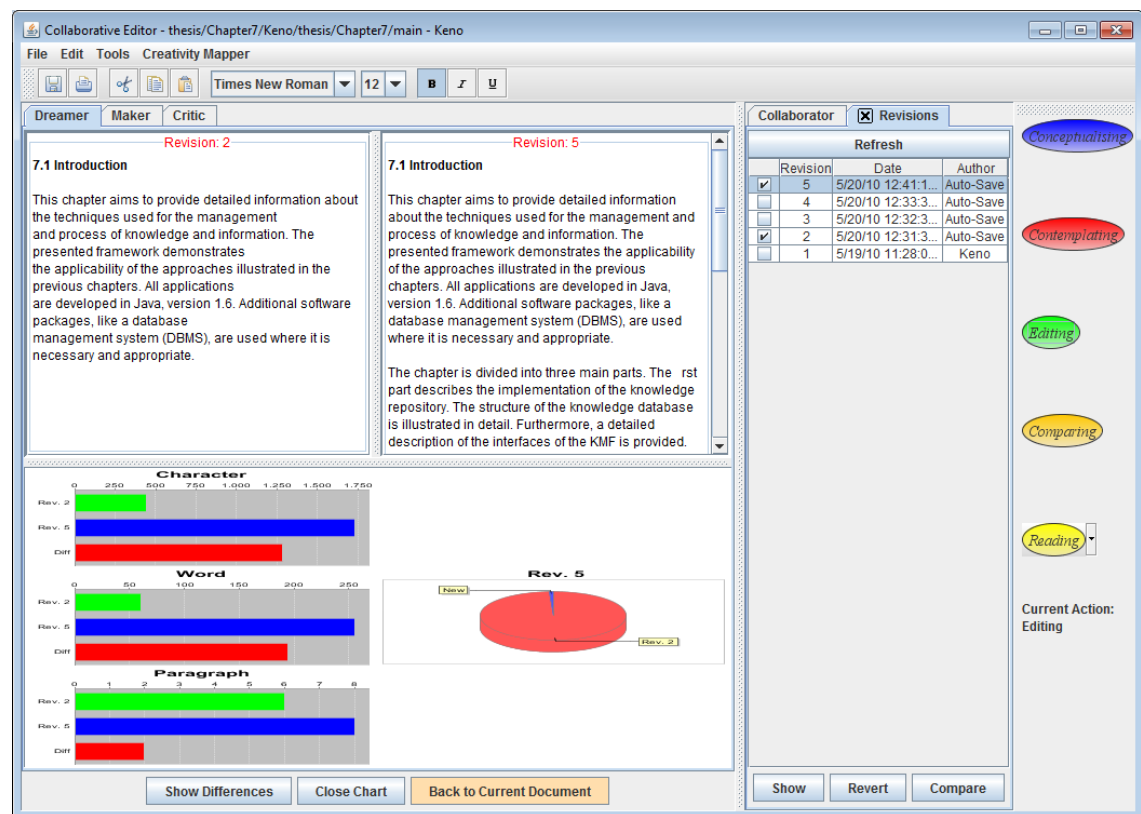


Figure 7.19: Collaborative Editor - Revision History

Figure 7.19 depicts the editor with the opened text analysis facilities window. The window shows the comparison of two versions of the same document. One can see the two versions next to each other as well as another pane with statistics about these two versions, below the text panes. Another new pane, listing the version of the document, is located on the

right. Furthermore, this pane contains three buttons: *Show*, *Revert* and *Compare*. The functions of this buttons are very much self-explaining and should not require and further description.

With this facility, the writer is able to analyse his writing progress through several different graphics, presenting detailed information. It also provides a good solution for restoring a previous version of the same document.

7.6.4 Supporting Tools

The DMCE offers a number of tools for the support and management of projects. As a collaborative framework, the DMCE contains tools for the communication and user management.

A collaborative environment requires an access management to give rights to other collaborators for a document or project. The DMCA knows two different types of access: read only or read and write access. The owner can choose this setting during the invitation process. At the end, a process sends an invitation via email to the new collaborator.



Figure 7.20: Access Control Management Facility

The invitation dialogue is depicted in Figure 7.20. The dialogue provides a small form where the user will put the email-address, access rights and an additional message. The invited collaborator will receive an email with an invitation number, which must be entered

into a dialogue, provided by the project management facility.

The owner of a project can always change the access rights of the collaborators or remove them when needed. The dialogue for these settings is located on a second tabbed pane in the access control management facility.

Being a collaborative environment, the DMCE certainly offers an instant messaging facility. Furthermore, it combines messaging functionality with an integrated collaborative whiteboard for drawing e.g. sketches.

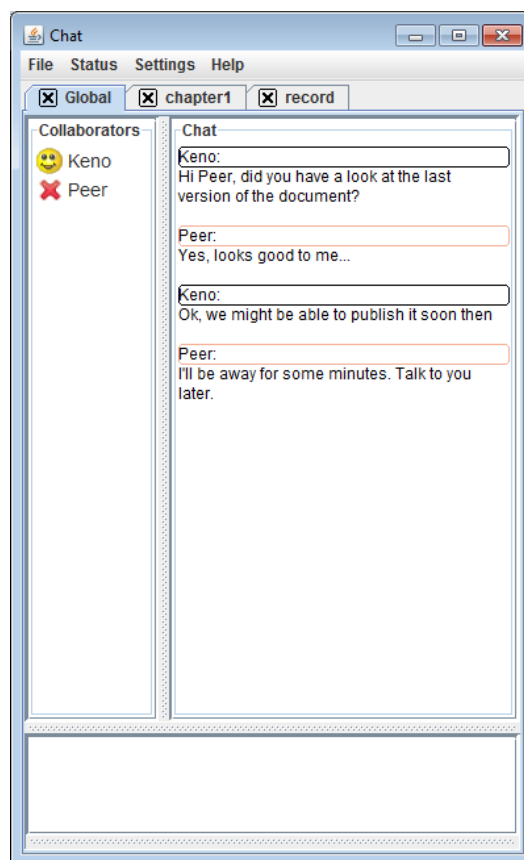


Figure 7.21: Instant Messaging Facility

Figure 7.21 depicts the instant messaging client of the DMCE. The client supports multiple chat channels. There are two different types of chat channels. There is one general channel, which every user of the system is able to join. The other type are private channels for the collaborators of a document. The user can open a chat channel for each document, which can be accessed through the project management facility.

The integrated collaborative whiteboard is not automatically opened. It is started via the menu: *File* → *Whiteboard*. The drawn image will be kept until it is deleted by a user. It is not automatically lost after closing the whiteboard.

Another supporting tool is called Pent-It Nodes. It is basically an electronic version of Post-It notes with an additional reminder functionality, based on a timer process.

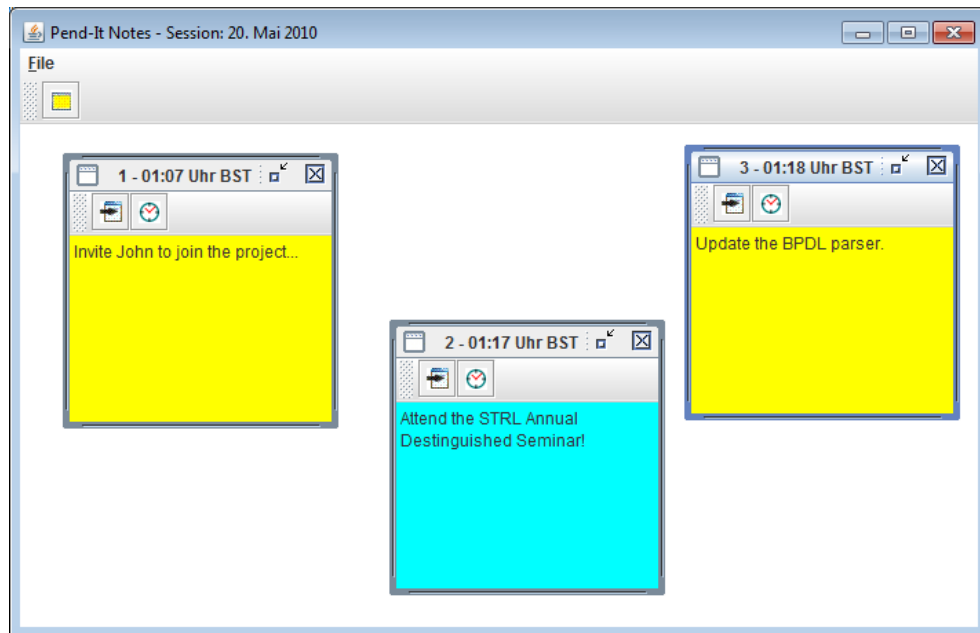


Figure 7.22: Pend-it Notes

The GUI of the application is somewhat similar to a pinboard. The window is mostly a white space, where the user can add new notes and move them around. A note consists of a smaller window with a small text pane and two buttons. The first button will send the note to the chat application. In this case, the text of the note is send to all members of the current active channel.

The second button will open a small dialogue to initiate the reminder function. There, the user can define the time for reminding him about note through a pop-up window.

The background colour can be changed. This enables a simple way of categorising the notes. The colour is changed by keeping the mouse button pressed on the note for at least a second.

7.7 De Montfort Creativity Mapper

Capturing the activities of the creator is a quite difficult task. The technique for the capturing process must be as precise as possible, but should not affect or even interfere with the creative process itself. The number of people which continue to use the DMCA after a first try is also depending on the acceptance of the capturing process. Creators will not accept any major interference with their work.

The De Montfort Creativity Mapper provides a simple and elegant technique for capturing the activities of the creator during the work on the artefact. The approach requires no special hardware for the input, but uses technology every computer user owns. This caters for the workplace-independency of the user. The system only requires a permanent internet connection during the working sessions.

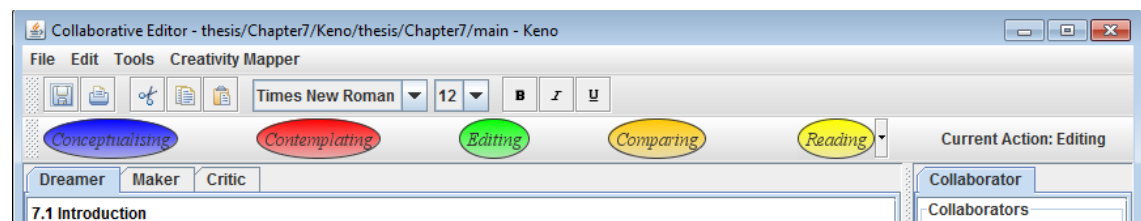


Figure 7.23: Creativity Mapper

The capturing process of the DMCM requires direct input from the user himself. Hence, it is important that the input can be provided quickly to avoid disrupting the work flow of the user. A good solution is the usage of buttons. A click on a button is not very time-consuming and becomes natural after a certain time.

A depiction of the DMCM is provided in Figure 7.23. The design of the buttons is different to the normal buttons of the DMCE. Being oval-shaped and colour-coded makes them easy to locate without the need for a closer look.

The pane with the buttons can be dragged to almost every position of the window of the collaborative editor. It is even possible to drag the pane out of the editor window. This provides more flexibility for the user.

A click on a button will not trigger any visual actions, besides displaying the new current action on the creativity mapper pane.

The creativity mapper can be customised to fit the requirements of the project. It is possible to define new viewpoints as well as new activities. Already existing entries can be changed or removed. The window with the settings for the creativity mapper is started via the menu entry *Creativity Mapper* → *Configure Layout*. Furthermore, it is possible to store the settings in a file, enabling the export of these.

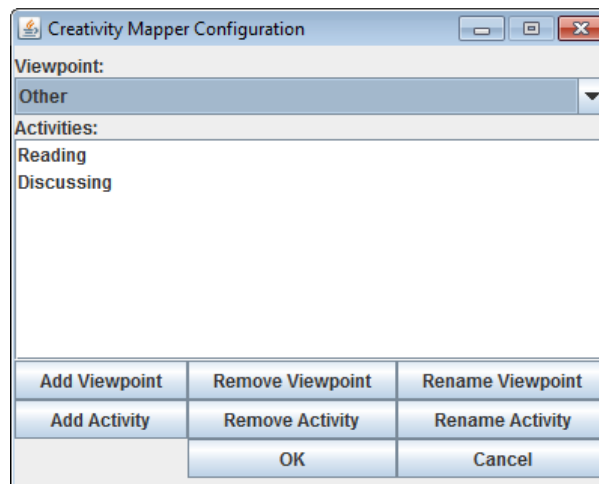


Figure 7.24: Creativity Mapper Configuration Dialogue

The window of the configuration is depicted in Figure 7.24. It contains buttons for adding, removing and renaming viewpoints and activities. The creativity mapper requires no further configuration than the here introduced features. It is an easy to learn application, usable without any long adaptation phase.

7.7.1 Creativity Map Viewer

The collection of the data for the creative process is fully transparent for the user of the system. The user can have a look at the creativity map through the visualisation tool of the DMCM. The creativity map is updated in real-time.

A small viewer application enables the creator to have a look at the creativity map for a project. This application is very useful e.g. if the creator likes to review the last activities of the creative process. The application can be opened via the menu entry *Creativity Map* → *Show Creativity Map*.

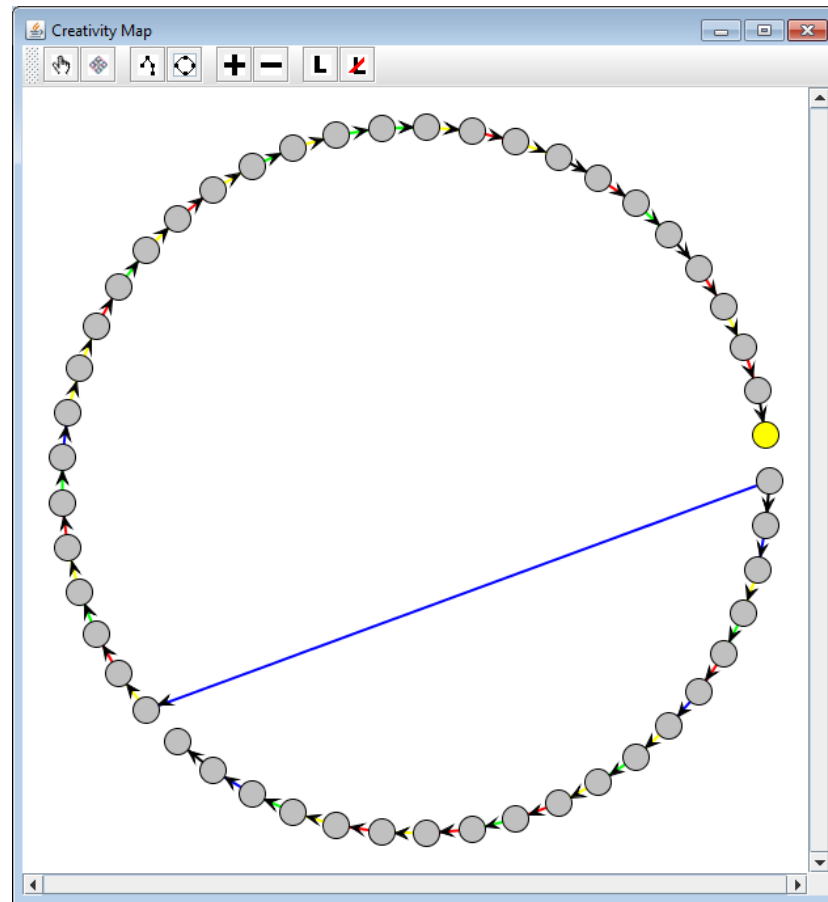


Figure 7.25: Creativity Map Viewer

The viewer application is depicted in the figure above. It is basically a huge canvas for depicting the creativity map and a toolbar with buttons for the modification of the view. The latest state is highlighted through a yellow coloured circle, instead of the standard grey. The user can choose between two different layouts for the creativity map. The first is the tree layout, which is the usual choice for depicting the creative process. The second layout type is depicted in the figure, the circle layout.

The application is also allows zooming into the map and moving states and transitions. Zooming is very helpful for the inspection of large creativity maps.

The labels of the transitions are hidden by default, but the user can enable displaying them. The transitions are always coloured, matching the viewpoint. The buttons of the creativity mapper have the same colour as well.

7.8 Knowledge Management and Analysis Tool

The last part of this chapter presents the initial design of an application, which enables the user to handle the knowledge management and the analysis of the creativity maps. The tools represent the implementation for the in this thesis introduced information extraction and knowledge management framework.

The knowledge management tool is used to add, change or delete the entries of the knowledge database. The application provides three different lists. The first list enables the access to the personal behavioural patterns of the user. The second and third list contain behavioural patterns of the domain and common category, which were selected by the user.

A form enables the user to set the following information of a personal behavioural pattern:

- Name of the pattern
- The pattern expression
- A descriptive text
- Keywords for a search and categorisation
- The type of application
- An optional message
- An optional constraint

This information represents the elements of a knowledge entry as described in Chapter 5. Additionally, each entry of the knowledge database carries a name for a better distinction. The information set represents all required information for the description of a behavioural pattern.

The analysis tool enables the extraction of information from the creativity map through behavioural patterns. The user is able to choose a behavioural pattern from a list, which is then applied on the creativity map. The results of the identification process are displayed in a graphical manner.

The two applications are located in the same application window. The window makes use of tabbed-panes for quick switches between the two displays. The two applications are not

part of the creativity map viewer in order to be able to provide both features at the same time. Future versions of the tool might use this for additional features, like highlighting the results of the identification process within the creativity map itself.

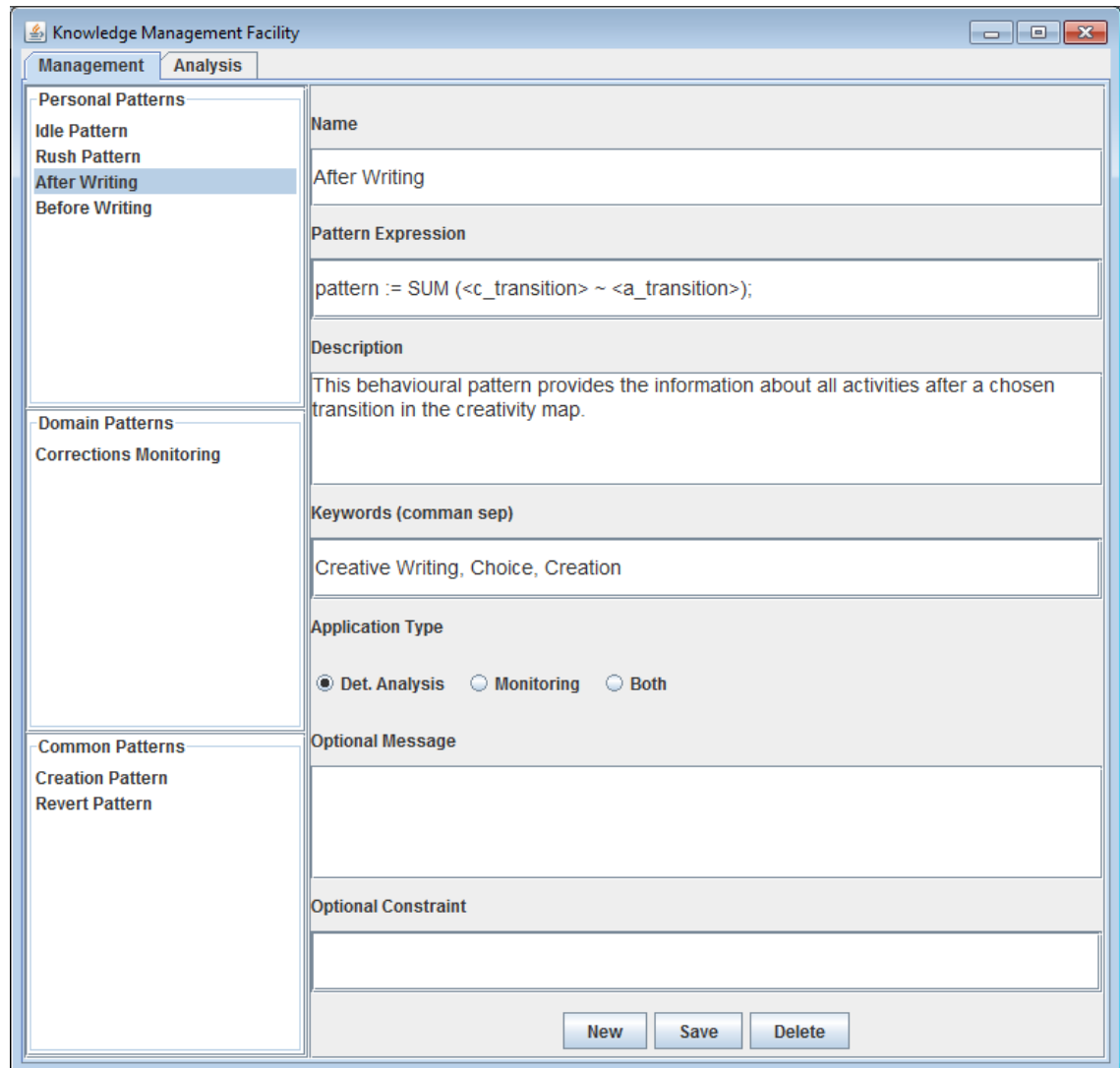


Figure 7.26: Knowledge Management

Figure 7.26 depicts the initial design for the knowledge management tool. The three different lists, containing the behavioural patterns, are located on the left side of the window. The settings itself are located on the right side. Buttons at the bottom of the window enable to save changes, create a new entry and delete an entry.

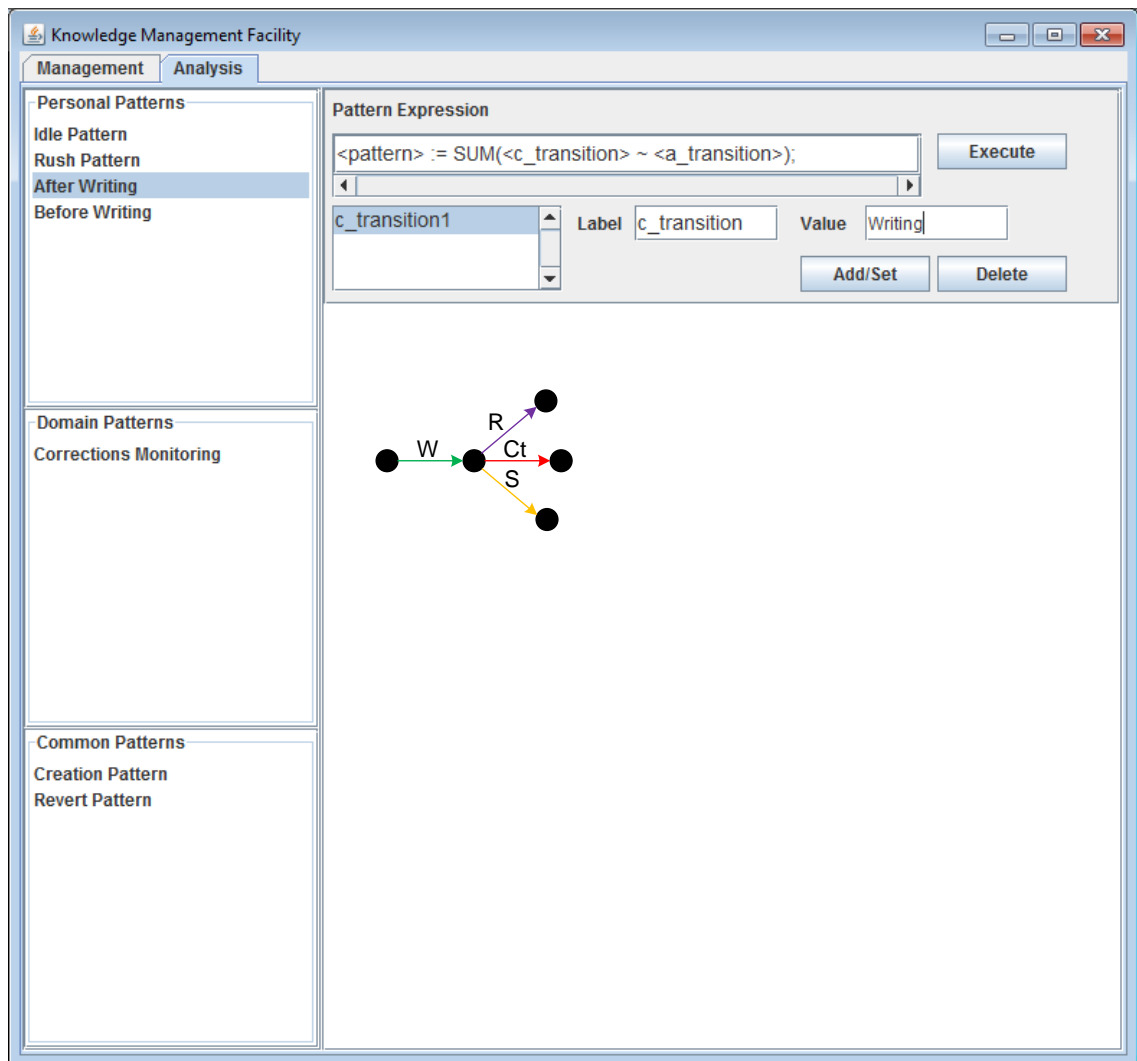


Figure 7.27: Analysis Facility

The analysis facility, depicted in Figure 7.27, is accessed through the tab with the label “Analysis”. The window consists of three main areas. A list with the behavioural patterns of the user is located on the left side of the window. The second area of the window contains details of the selected behavioural pattern. The information includes the pattern expression, text fields for the definition of the chosen-transitions. Furthermore, the area contains a button for the execution of the behavioural pattern. A click on this button will initiate the parsing process.

The results of the parsing process are depicted below the “definition area”. All results are depicted as a graphical representation. The concept of the BPDFL allows a quick modification of the pattern expression. This enables interactive analysis processes.

7.9 Summary

The De Montfort Creativity Assistant (DMCA) and its applications have been developed for the support and analysis of creative processes. This software framework features a pluggable design, which caters for a collaborative, domain-independent creativity support. The collection of data about the creative process is vital for a later analysis. Therefore, the aim of this tool set is not only the support of creativity for an individual creator, but moreover the collection of massive data to improve the analysis in general. It was mentioned in the chapters before, that the analysis and therefore the understanding of the creative process will grow with the amount of data, which is available for analysis. The collection of data and the support of collaboration in particular make the De Montfort Creativity Assistant a very valuable source of data for this task.

The chapter continues with the introduction of techniques used for the Behavioural Pattern Description Language (BPDFL) parser as well as techniques for the traversing of creativity maps. The approaches provide the functionality as described in the previous chapters. The focus lies on the algorithms and techniques behind the analysis of a creativity map. The introduction starts with the description of a technique for the identification and storage of revert sequences. This is followed by a detailed, step-by-step explanation of the BPDFL parser implementation. It contains information about the architecture of the parser, followed by an explanation about how the parser is used with the creativity maps. The algorithm of the monitoring process is explained via a flow chart.

The last part of the chapter introduces the tools of the De Montfort Creative Environment (DMCE) and De Montfort Creativity Mapper (DMCM), which together represent the client-side of the DMCA. The DMCE provides a tool-set for the creator, enabling him to work collaboratively on an artefact. The current version of the DMCE contains a collaborative editor, a project management facility, a chat facility with integrated whiteboard and a tool for taking notes. The idea behind this tool-set is the creation of a complete, but freely adjustable working environment for creators.

The De Montfort Creativity Mapper represents the interface for the collection process of the creator's activities. The simple GUI allows the creator to directly start working, without a long adaptation phase. Furthermore, the definition of the activities is fully customisable. The creator can quickly adjust the set of activities as well as change the granularity of the activities.

An initial version of a knowledge management and analysis tools is described in the section before. It provides the interfaces for the KMF as well as the facilities for the analysis of

the creativity maps. These two interfaces have been coupled in order to enable the user to make quick changes to the knowledge entries or a quick application of newly defined behavioural patterns on a creativity map.

The implementation of the De Montfort Creativity Assistant and its tools shows that the presented approach is not only a theoretical model, but has already been tested and used for the case studies, which are presented in the following chapter. Details about the implementation such as Unified Modelling Language (UML) diagrams can be found in the appendix of this thesis.

Chapter 8

Case Studies

Objectives

- Exploration of the creative process of collaborative writers.
- Exploration of the creative process of the author when writing this thesis.
- Exploration of the creative process of a musician.

“Research is the process of going up alleys to see if they are blind.”

Marston Bates (1906-1974)

8.1 Introduction

This chapter presents the results of three selected experiments, which were conducted in order to test the framework itself as well as to study the creative process of creators from different domains. The aim of the case studies was a test of the approach under real working conditions, in order to gain precise information about its applicability.

The data, collected in these experiments, is provided in form of the creativity maps in the appendix of this thesis. The creativity maps are not shown in this chapter because of their large size.

8.2 Case Study 1 - Participants of a Writing Workshop

This case study is based on the results of a creative writing workshop with the De Montfort Creativity Assistant. A number of experiments were conducted in order to gain information about the creative behaviour of the participants as well as to study the acceptance of the De Montfort Creativity Mapper for the capturing of the creative process.

The following paragraphs provide some general details about the workshop.

Venue: 30th July 2009, STRL, Gateway House, De Montfort University

Participants:

Number of Participants: 9

Age: 23 - 60

Occupation: Students, University Staff, Employed outside the University

Duration: 9.30am - 12.30am

Setup: Every person worked with a separate PC. The software testing was split up into three parts. A discussion about the tool and the event took place at the end of the workshop.

Part 1: Start and login into the environment. Followed by writing a short text about themselves in order to get used to the software. (10 - 15 minutes)

Part 2: Writing of a short text about a stimulus item (e.g. watch, necklace, sunglasses, etc.), provided by the leader of the workshop. There was only one exemplar of each item. This induced the participants to write different stories. (ca. 25 minutes)

Break (20 minutes)

Part 3: Editing of the written text to reduce the number of words by half. The idea of this part was to induce the writer to review the text, which would possibly cause several revert activities. (ca. 25 minutes)

The final discussion went on for more than an hour. The participants were asked to tell their view about the approach and the tool-set. Furthermore, to talk about any problems occurred with the software and possible improvements.

The concept of the approach was generally very well accepted. The De Montfort Creativity Mapper seems to fully satisfy the requirements for capturing the creative process. Suggestions have been made about additional shortcuts for the selection of an activity.

8.2.1 Comparison of Creative Behaviour

Having several creativity maps from writers working on the same topic is very beneficial for a comparison of the creative behaviour. The short time span of the experiment is helping to directly compare the activities without further consideration of their position inside the creativity map. This would be necessary in a larger creativity map as the requirements for certain activities change as the work progresses.

Creation Pattern

The first analysis is based on a creational behavioural pattern. The aim of the pattern is the identification of the activities, responsible for changes to the artefact. The expression for the used behavioural pattern is:

$$\langle pattern \rangle := SUM(\{ \langle a_transition \rangle \setminus (Editing) \} - \langle c_transition \rangle);$$

This expression will cause the parser to retrieve the longest possible sequence in front of every *Editing* transition, which does not contain an *Editing* transition itself. The results are finally summarised into one entity to provide a better overview.

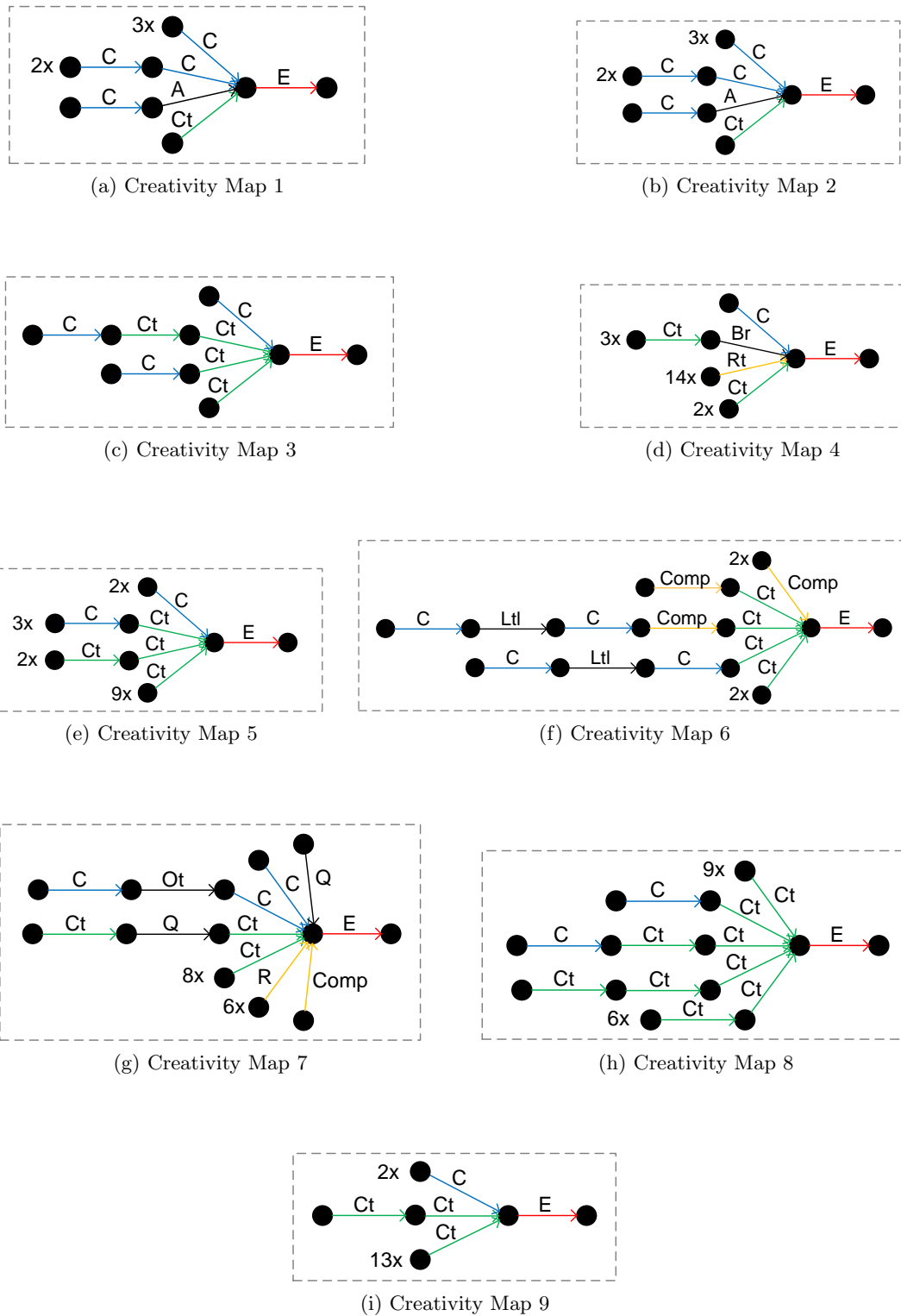


Figure 8.1: Results for the Creation Pattern












Action	Description
	Editing
	Contemplating
	Conceptualising
	Comparing
	Reading through
	Reading
	Break
	Listen to instructions
	Awaiting
	Questioning
	Other

Figure 8.2: Description of the Labels

Figure 8.1 contains the results of the parsing process. The black states represent a placeholder and not a specific state itself, as described in Chapter 4. A description of the labels is depicted in Figure 8.2. It is common that a sequence or path is often reoccurring within the creativity map. Multiple occurrences of a path are marked in the result (e.g. 2x). Paths, which only occur once, have no information about the frequency.

The process has produced some interesting results with very different entities for each participant. The first difference, one will notice, is the number of activities, finally leading to the creational (*Editing*) activity. Some writers have a very short sequence between two creational activities. This is clearly visible in the creativity maps number two, four and nine. Other writers, however, have a longer sequence between the creational activities. The creator of creativity map number six has sequences with up to six activities (including the *Editing* transition).

Another interesting discovery is the frequency of how often a sequence or path is used. One half of the writers have a balanced number for all sequences. The other half, however, has clearly a preference for one or two sequences. These writers could use the results to further analyse their creativity by inspecting the changes made to the text after a frequent sequence. It might indicate very creative behaviour or poor creative behaviour. In the latter case, it would be a good idea to create a behavioural pattern for the monitoring process in order to inform the user about this sequence.

Revert Pattern

In the second part of the analysis, the focus will be on situations where the writers went back to an older stage of the project. This behaviour indicates that the writer was stuck in his last stage, which might be the result of poor creative behaviour. This phenomenon should provide some interesting information.

The collection of this information is only possible with the help of the *REVERT()* operator. A parameter will not be used in order to retrieve all revert sequences.

The BPDFL expression of the pattern is the following.

$\langle pattern \rangle := REVERT();$

The identification process retrieved following elements for the behavioural pattern.

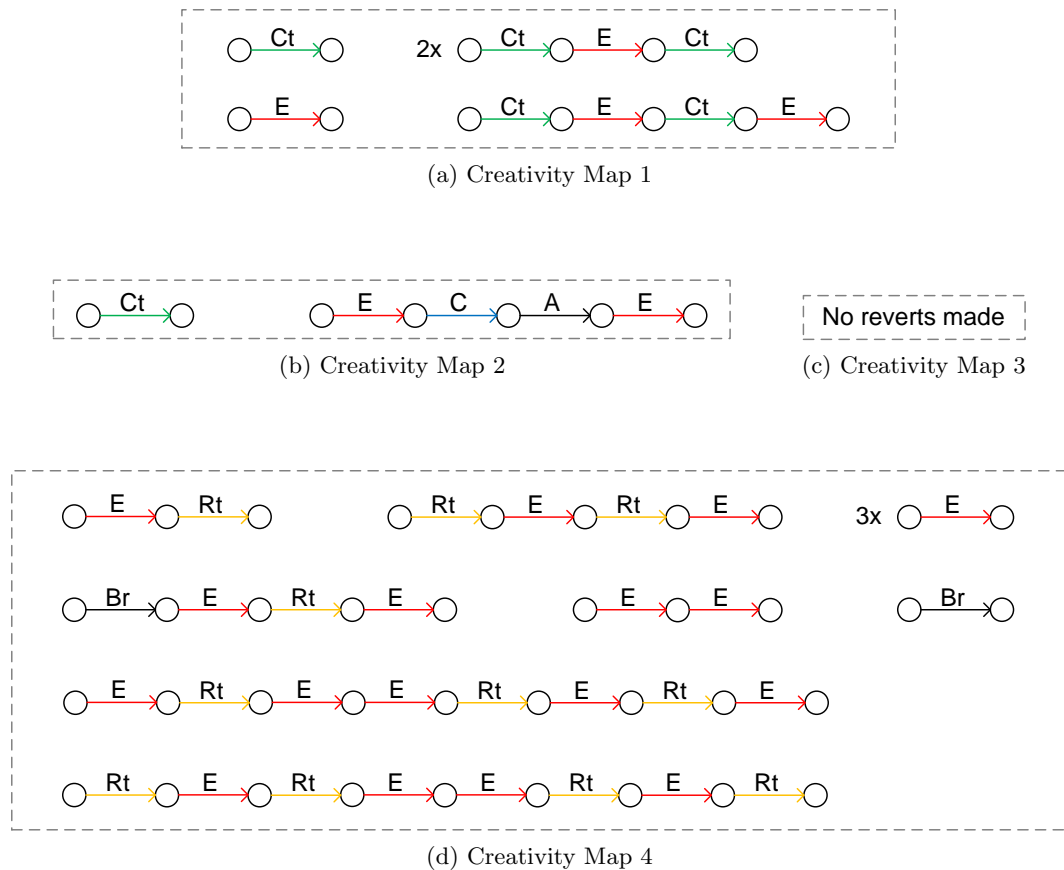
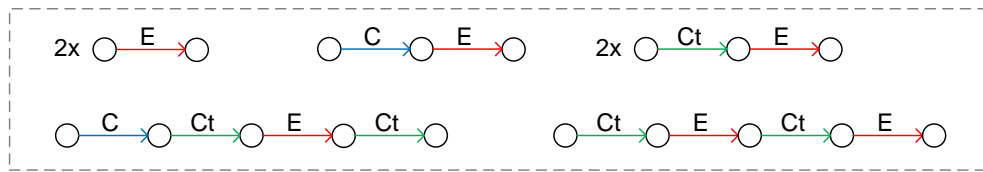
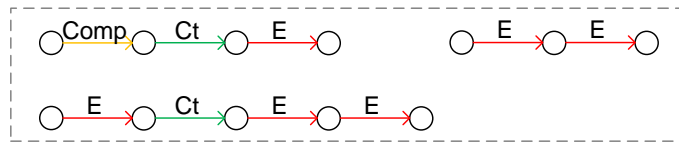


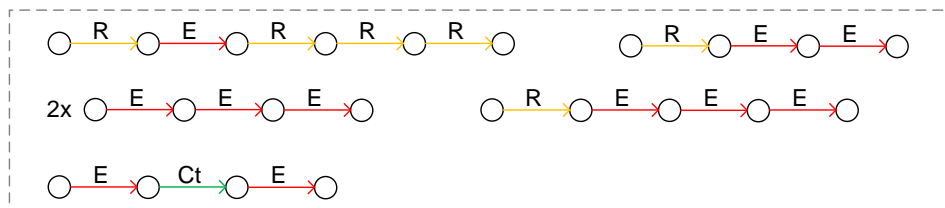
Figure 8.3: Results for the Revert Pattern - Part 1



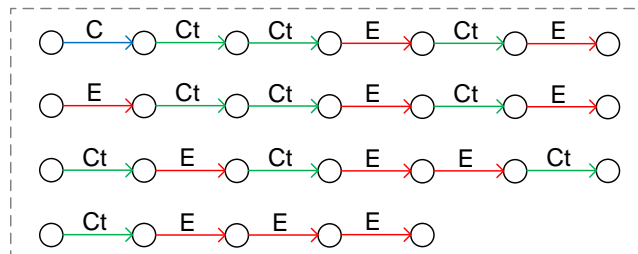
(a) Creativity Map 5



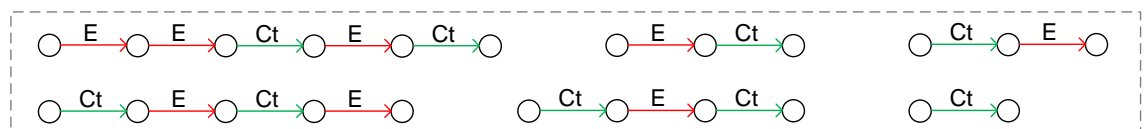
(b) Creativity Map 6



(c) Creativity Map 7



(d) Creativity Map 8



(e) Creativity Map 9

Figure 8.4: Results for the Revert Pattern - Part 2

Figure 8.3 and 8.4 depict the results of the identification process. The writers showed differences in their creative behaviour regarding the need for reverting to a previous state. One of the participants showed a rather odd behaviour in his creativity map. The map

itself contains no revert operation. The reason for this could be that the participants did not make corrections to the written text. A further analysis of the timestamps could be interesting in order to look for a possible false usage of the DMCM.

One interesting discovery of the analysis is the knowledge that a number of revert sequences reoccurred in some creativity maps. This is very interesting in regard to the identification of poor creative behaviour, which has a negative influence on the creation process. Furthermore, the revert sequences are often very similar, containing the same activities. Examples are sequences with combinations of just the two *Contemplating* and *Editing* activities. Those activities can be of course the dominant activities for the creation process itself. Nonetheless, the analysis shows that some of these combinations almost exclusively lead to a revert operation for some of the writers.

The results for creativity map number four have one interesting attribute. None of the result entities contains a *Contemplating* activity. It seems that this fact indicates a negative influence on the creative behaviour of the participant. This type of behaviour did not occur in any other creativity map for this workshop.

8.2.2 Conclusion

This case study is a good example of how the creative behaviours and the creative processes of creators can be compared with creativity maps. The creativity maps are rather small due to the time limit of the workshop. Nonetheless, the analysis results provide several interesting information about the participants.

The most interesting creativity map was created by participant number seven. The map contains a large number of revert operations. The sequence of activities, which lead to the revert operation, contain some peculiar repetitions of *Editing* and *Reading* activities. The high frequency of revert activities is closely related to a high frequency of the sequence $S = \rightarrow_R ; \rightarrow_E$, identified through the creation pattern, depicted in Figure 8.1g. This information allows the definition of a new behavioural pattern for the monitoring process. Through this, the participant would be informed about this type of behaviour and could be guided to undertake another type of activity.

The interpretation of the analysis results shows that behavioural patterns are most useful when used in a set. This enables the best possible understanding of the retrieved information and is also related to the the cycle of analysis and perception, which was already discussed in Section 7 of Chapter 6.

8.3 Case Study 2 - Self-Observation

The second case study in this chapter discusses the analysis of two creativity maps, representing the creative process during the creation of the second and eighth chapter of this thesis. The aim of this case study is a self-observation of the creative process of the writer of this thesis himself.

The creativity maps are not representing the complete creative process for each chapter, but a large part of it. An analysis of the two creativity maps will help to investigate the differences in the creative behaviour between the two chapters. The question is how much the behaviour has changed between the chapters. The two chapters contain different types of content. Chapter 2 contains a summary of papers, which requires an extensive search for information. Chapter 7, however, contains an introduction of a software application. The focus changed from the search for information to the adequate presentation of the software. The different requirements for the chapters will certainly have an effect on the behaviour of the writer.

The creativity maps for this case study can be again found in the appendix of the thesis.

8.3.1 Analysis of Changes in Creative Behaviour

The aim of this case study is to find out how different requirements for similar tasks influence the creative process of the same person. Both of the here presented creativity maps were created during writing sessions. The question is, whether the creative behaviour did change and if so, how it has changed. A number of behavioural patterns will be applied on the two creativity maps in order to retrieve meaningful information.

Choice Pattern

The first extraction process will collect the information about the first activity after each *Editing* activity. A so-called choice pattern will be used for this extraction process. The aim of the application of this pattern is the collection of information about the number of occurrence for each activity. This will provide an insight about the first step after the creation. The focus lies on a possible change in the requirement for new information input through *Reading* and *Searching* activities.

Following BPDFL expression will retrieve the information.

$$\langle pattern \rangle := SUM(\langle c-transition \rangle \sim \langle a-transition \rangle);$$

The pattern expression is quite simple. The chosen transition is followed by the branching operator (\sim) and an any-transition. The any-transition is not enclosed by curved brackets. This limits the sequences to a length of two elements. The $SUM()$ operator will summarise all result entities into one element.



Figure 8.5: Activities after Editing






Action	Description
 E	Editing
 Ct	Contemplating
 C	Conceptualising
 S	Searching
 R	Reading

Figure 8.6: Description of the Labels

Figure 8.5 depicts the results of the information extraction process. There are indeed differences in the creative behaviour. In Chapter 2, there was an almost exclusive need for new information through the *Reading* and *Searching* activities. In Chapter 7, one can see a much more balanced need for the particular activities. The need for *Reading* activities in Chapter 2 fits perfectly to the objectives of the chapter itself. In opposite, Chapter 7 requires no intensive search for information, because the described tool was developed by the author himself.

The description of the labels is depicted in the table of Figure 8.6.

Revert Pattern

The next step of the analysis will provide a look into the revert sequences. After being able to identify differences in the creative process, it will be interesting to see whether this has also an effect on the occasions where the author got stuck with his creative flow.

The same BPDFL expression as in the first case study will be used.

$\langle pattern \rangle := REVERT();$

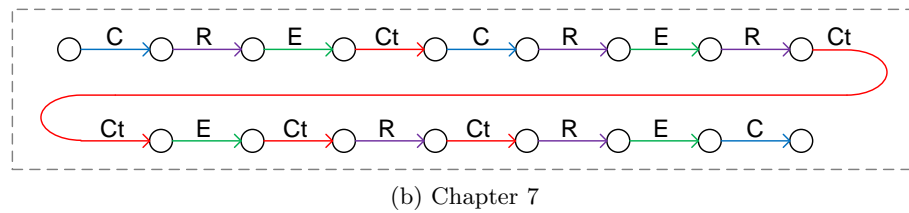
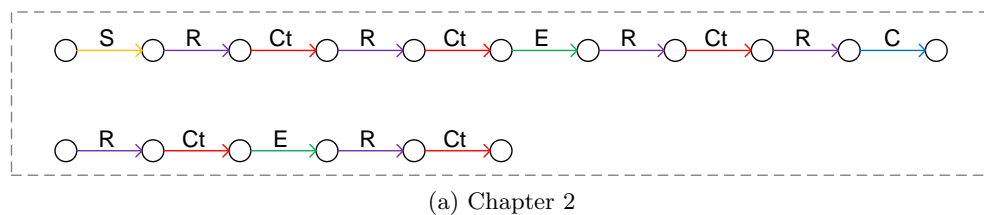


Figure 8.7: Results for the Revert Pattern

The results of the identification process, depicted in Figure 8.7, are different than the results of the first case study. There are not many revert sequences in the two creativity maps. The sequences are quite long compared to the results of the ones from the first case study.

The comparison of the revert sequences of the two creativity maps does not show any fundamental differences. Nonetheless, there are differences in the positions of the sequences. The revert operations of the creativity map for Chapter 2 were made at the beginning of the creative process. The revert operation of the creativity map for Chapter 7 was made in the middle of the creative process. It also leads to a complete restart from scratch. Hence, there must have been a bigger problem with the initial concept of Chapter 7.

Contemplating/Editing Pattern

The next part of the analysis will be based on knowledge, gained through the first case study. The analysis of the revert sequences for the creativity maps of the first case study showed some similarities regarding combinations of *Contemplating* and *Editing* activities. Sequences containing mostly these two activities were often the reason for a revert operation. Hence, one can assume that this behaviour has a bad influence on the creative process of writers. It might be a domain-specific pattern.

A behavioural pattern will help to identify this type of sequences in the creativity map of writer of this thesis. All creativity maps belong to writing projects. The following BPDFL expression describes the behavioural pattern.

$\langle pattern \rangle := LEN(\{\langle c_transition1 \rangle \mid \langle c_transition2 \rangle\}) (> 2);$

with $\langle c_transition1 \rangle = \textit{Contemplating}$ and $\langle c_transition2 \rangle = \textit{Editing}$

The expression will instruct the parser to search for sequences with only *Contemplating* and *Editing* activities, with a length of more than two transitions/activities.

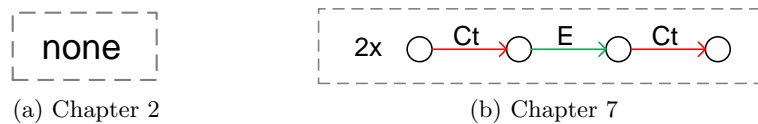


Figure 8.8: Results for the Contemplating/Editing Pattern

The result set of the identification process was small. Figure 8.8 depicts the entities. No matches were found in the creativity map of Chapter 2. The creativity map for Chapter 7, however, contains the pattern twice. There are no indications that the two appearances of that pattern were responsible for a revert operation.

It is possible that this type of behaviour does not negatively influence the creativity of the writer. Nonetheless, both creativity maps contain only a few revert operations. Perhaps it is necessary to analyse more data in order to make definite statements about the creative behaviour of the writer.

8.3.2 Conclusion

The analysis of the creativity maps has shown that the creative behaviour changes during the work on a larger project. It confirms the general assumption that creative processes are very specific for a certain task. The comparison of the results with those of the first case study reveals that people can have different approaches on how to process a similar task.

Furthermore, the analysis provided an example of how the knowledge, gained through other projects, can be used for future analyses. It is easy to define a new behavioural pattern, based on the knowledge from previous analyses. The management and store of this knowledge is the task of the knowledge management facility. It is obvious that the knowledge from one analysis is very beneficial for other analyses.

8.4 Case Study 3 - The Musician

The last case study of this chapter describes the analysis of a special creativity map. The data for the creativity map was not collected through the De Montfort Creativity Mapper. Instead, it is based on secondary input material. Secondary means that this data was not intended to be used for the creation of a creativity map. The material is a voice recording of a musician, who is composing a new piece of music. The recording represents a one hour session, consisting of verbal explanations, playing music and large parts of composing activities.

The creativity map for this project relies on the identification of the activities, the musician had undertaken. There is not particular explanation of the undertaken activities. One of the main questions for this case study is to find out whether it is possible to reproduce the creative process and to create an exact creativity map for the session. Furthermore, to find out if the creativity map contains enough information about the creative process, in order to enable a detailed analysis, based on behavioural patterns.

A first attempt regarding the above mentioned points has been already described in the paper Mapping Human Creativity [136]. The details of the experiment can be found in Section 6.1 of the paper. It describes the creation of a creativity map and its analysis for the same project. This case study continues the research with new data and new, advanced analysis techniques.

8.4.1 Background

The music, which was created during the recorded session, is part of a large project, called the Secret Garden. The Secret Garden is a multimedia project in collaboration with the Institute of Creative Technologies (IOCT). It includes music, video and digital environments.

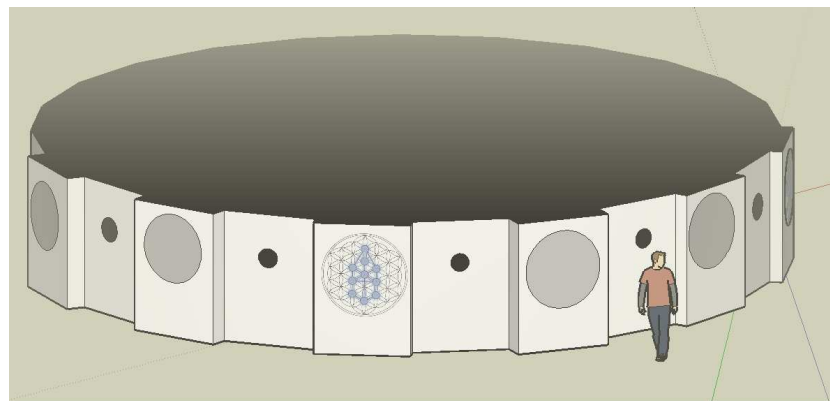


Figure 8.9: Secret Garden

Figure 8.9 illustrates the Secret Garden with the view from outside. It is basically a cylinder with a number of windows. A look into one of these windows will show an idyllic scene of the Garden Eden. There are eleven windows in total and each will show a different scene, accompanied with music. The members of the project describe the Secret Gardens with following words.

Secret Garden is an attempt to recreate a contemporary version of Eden in the midst of an urban space using the stories of the Fall. Two human figures dressed from head to toe in robes of blue (Male) and red (Female) will enact the story of the Fall, combining video vignettes with 3D generated environments, each scene distributed to a different viewpoint. Each viewing port will contain two loudspeakers, through which will be heard settings of eleven texts for solo voices (one male, one female).

As mentioned before, the analysis will focus on the music part of the project. The creation of the three-dimensional environments is not included. The creation of the creativity map itself will be again not part of the case studies. However, some comments about the creation process will be made in between.

8.4.2 Analysis of the Creativity Map

The main objective of the last case study is to find out whether the creativity map contains enough information for an analysis. Firstly, the sequences, which lead to a creational activity, will be analysed. It will be interesting to search for anomalies in regard to normal creativity maps. This could be for example very short sequences - a hint for insufficient input data. Finally, the creative process will be analysed in order to search for interesting information about the creative behaviour of the musician.

Creation Pattern

The Creation Pattern is the first pattern which will be applied on the creativity map. This is a good exercise for an inspection of the sequences, which lead to an creational activity.

$$\langle pattern \rangle := \langle a_transition \rangle \setminus (\text{"Composing"}) \\ \langle a_transition \rangle \setminus (\text{"Composing"}) \langle c_transition \rangle;$$

with $\langle c_transition \rangle = \text{Composing}$

The creational activity for this creativity map is *Composing*. Hence, the parser will extract sequences, which have this type of transition at the last position.

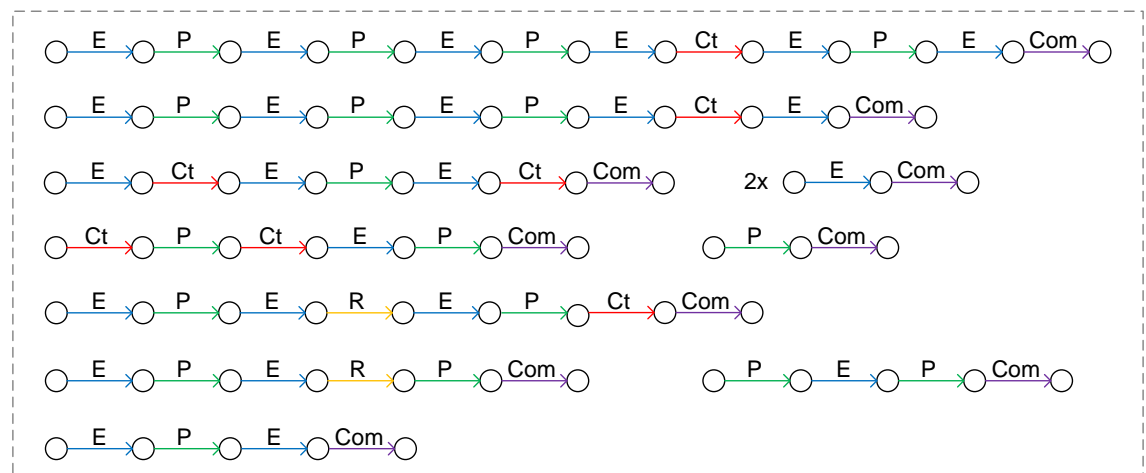


Figure 8.10: Results for the Creation Pattern

The identification process retrieved quite long sequences with up to twelve transitions. Every entity of the result set contains only one creational activity due to the *without* operator (\setminus (“Composing”). The descriptions for the transitions can be as always found in the table below.

Action	Description
\xrightarrow{E}	Explaining
\xrightarrow{Ct}	Contemplating
\xrightarrow{P}	Playing
\xrightarrow{Com}	Composing
\xrightarrow{R}	Reading

Figure 8.11: Description of the Labels

One will notice a dominance of *Explaining* and *Playing* activities in the result set. These two activities covered the majority of all activities. The following behavioural pattern will extract all sequences, consisting only of those two transitions.

$\langle pattern \rangle := LEN(\{ \langle c_transition1 \rangle \mid \langle c_transition2 \rangle \}) (> 1)$;

with $\langle c_transition1 \rangle = Explaining$ and $\langle c_transition2 \rangle = Playing$

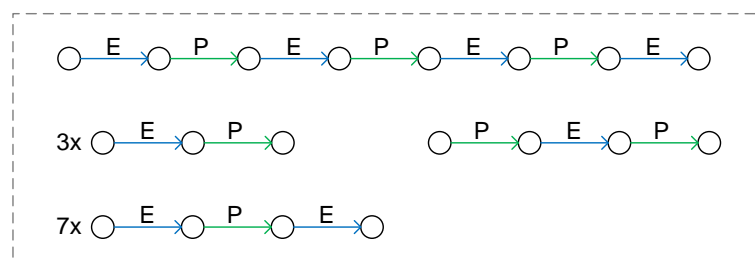


Figure 8.12: Results for the Explaining/Playing Pattern

Figure 8.12 depicts the result set for the behavioural pattern. There are two sequences in particular, which cover most cases. However, if one remembers the fact that the creativity map is based on audio file, recorded with a dictaphone, it might be necessary to ignore the *Explaining* activity. This activity is not involved in the creation of the new song itself, but focussing on recording process. Hence, it is similar to the activity of pressing the buttons

of the DMCM, when recording the activities in the usual way. The activity itself has no effect on the creation process.

As a result, it is very helpful to create a partial creativity map in which the *Explaining* activity is hidden. This partial creativity map enables the analysis of the important activities.

8.4.3 Analysis of the Partial Creativity Map

The analysis of the partial creativity map follows the same rules as used for the normal creativity maps. The first part of the analysis will again look at the sequences, which lead to a creational activity.

Creation Pattern

The used behavioural pattern is the same as before.

$$\begin{aligned} \langle pattern \rangle &:= \langle a_transition \rangle \setminus (\text{"Composing"}) \\ &\quad \langle a_transition \rangle \setminus (\text{"Composing"}) \langle c_transition \rangle; \end{aligned}$$

with $\langle c_transition \rangle = Composing$

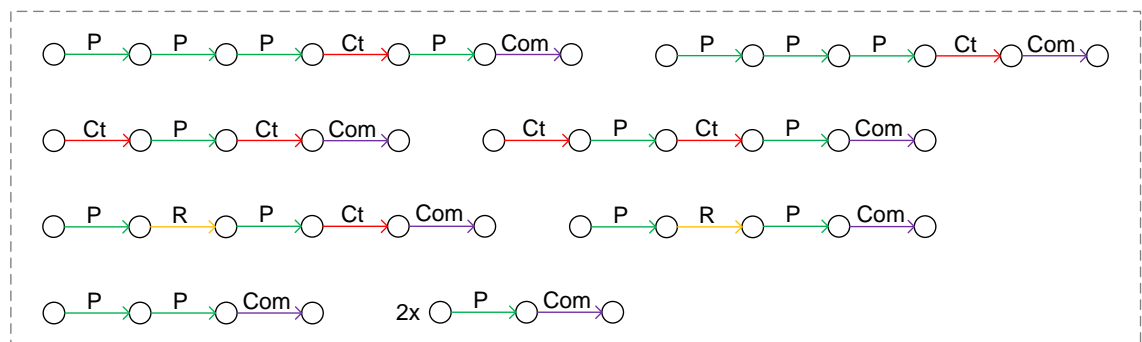


Figure 8.13: Results for the Creation Pattern

Figure 8.13 depicts the results of the identification process. The entities are basically the same sequences as before, only without the *Explaining* activities. The two \rightarrow_E ; \rightarrow_{Com}

sequences are of course missing for this result set. One could now argue that this partial creativity map will lead to false conclusions about the creative process, as all information is important. However, the *Explaining* activity before the creational activity does not necessarily mean that it lead to the creation. It could be, and this is much more likely, a pause during the *Composing* activity. The reason for this assumption is the fact that there was always another *Composing* activity before the *Explaining*.

Choice Pattern

The last part of the analysis will focus on the probability the user will choose a certain activity. This information can be retrieved with the help of a choice pattern.

$$\langle pattern \rangle := SUM(\langle c.transition \rangle \sim \langle a.transition \rangle);$$

with $\langle c.transition \rangle =$ the subtitle of the figures below.

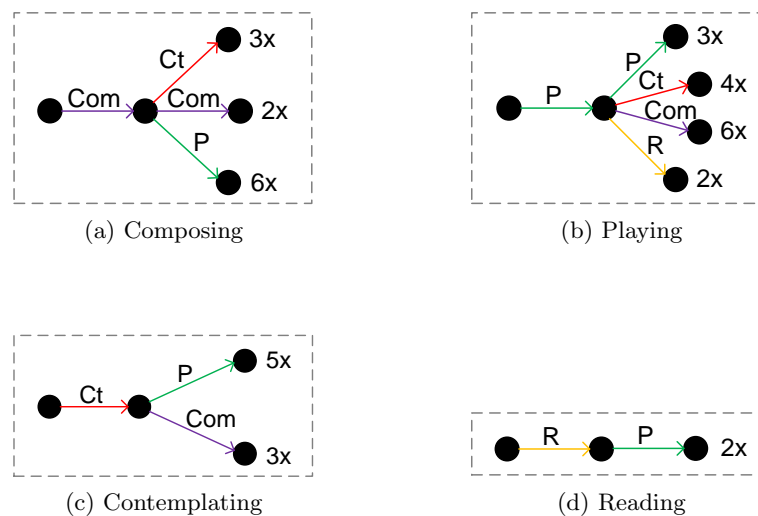


Figure 8.14: Results for the Choice Pattern

The result set of the identification process is depicted in Figure 8.14. It is obvious that the musician has no special preference for choosing one or more activities after the given one. However, this only correct for the number of occurrences. The musician makes use of all his activities after *Playing*, but makes only use of one activity after a *Reading* activity. A similar behaviour can be found for the *Contemplating* activity.

Therefore, this analysis provides some very valuable information about the creative process. A next step for the analysis of the creative behaviour for this musician could be a session, focussing on the analysis of e.g. the *Contemplating* activity. It would be certainly interesting to see, whether the choice between the succeeding transition types makes a difference for the creative process. Perhaps the musician would benefit from using more than two different activities.

The creation of a partial creativity map was very beneficial for this case study. Without it, the *Explaining* activity would strongly influence the results of the analysis due to large number of occurrences. Nonetheless, partial creativity maps must be used carefully to avoid a loss of valuable information.

8.4.4 Conclusion

This case study was very useful for the demonstration that the in this thesis introduced technique for mapping the human creativity enables to retrace the steps of a person's creative process, based on secondary input material. The material can be either notes or voice recordings, like the one which was used in this case study.

An important issue for this type of creativity map is its detailedness. The creativity map can only be as detailed as its input material. Nonetheless, also a small amount of data contains very important information. Concerns regarding possible flaws in the order of the activities in the sequences were proven to be false. Instead, the order of the activities was very clear when listening to the recording.

The case study showed that the creativity map contains enough information for an analysis. The results for the behavioural patterns allowed to make assumptions about the creative behaviour of the musician. Especially the choice pattern could help to improve the creative process. Like in the previous case study, it became obvious that the analysis of the creative process becomes more detailed with every new step. The process of analysing is never finished, like creativity itself.

The creativity map and partial creativity map contain only one revert operation. It happened during a very early point of the session, the explanatory phase. Therefore, the revert operations have not been analysed as these seem neglectable. Listening to the recordings confirmed the impression. The working process of the musician was more or less straight forward.

8.5 Summary

In this chapter, it was shown how the, in this thesis introduced, techniques are used to analysis real data. The analyses presented some very interesting insights into the creative processes of the creators. Common creative behaviour was identified in the form of unwanted behaviour. As a result, it was possible to use this knowledge for the following analyses. The behavioural patterns and the Behavioural Pattern Description Language were successfully tested and turned out to satisfy the three requirements for an automated analysis of creativity maps: Simplicity, Flexibility and Reliability.

The first case study discusses the analysis of the creative processes of several participants of a creative writing workshop. The focus of this case study lies on the comparison of creative behaviours of the participants. All participants were undertaking the same task. This allowed to study the creative process and creative behaviour for similarities. The results of the analysis show that many of the participants share the same pattern when showing a poor creative behaviour. This enabled to creation of a behavioural pattern, which identifies this behaviour. It can be used in the monitoring process for preventing this type of behaviour in future.

A second case study is focussing on the creative process of the author of this thesis. The analysis consisted of the comparison of two creativity maps, which were created/recorded during the writing-up of Chapter 2 and Chapter 7. It was interesting to find out that the author had shown very different creative behaviours when writing-up those two chapters. Another interesting fact is the rare occasion of reverts to a previous state. This was a very different behaviour in comparison to the creativity maps of the first case study. The diversity of the undertaken activities in regard to their frequency appears to be very beneficial. It is worth compiling the knowledge about the creative behaviour of the author into a new behavioural pattern. As it would describe a sort of exemplary good creative behaviour.

The third case study presents a special case for mapping human creativity. The creativity map was created a posteriori, based on voice recordings made by the creator. Hence, the most important question to answer in this case study was to clarify whether the detailedness of the creativity map was sufficient for a detailed analysis.

The case study was a good example for the usefulness of PCM. It was necessary to hide a certain transition type as it was not part of the creation process itself but the voice recording.

One interesting analysis, the study of the effects of collaboration on the creative process, could not be presented in this thesis. The reason for this is the data requirements for this type of analysis. Analysing the effects of collaboration requires much more preparation and data than “normal” projects. It requires at least the creativity maps of all collaborators, working together on the same project as well as creativity maps when working alone. The two different types of creativity maps are necessary in order to be able to compare the creative processes. Nonetheless, the analysis of the effects of collaboration on the creative process is planned for the future work.

Chapter 9

Conclusion and Future Work

Objectives

- Drawing a conclusion about the presented work.
- Review of the used research methodology.
- Discussion of the limitations for the presented technologies.
- Discussion of possible future work.

“Enough research will tend to support your conclusions.”

Arthur Bloch (*1948)

9.1 Introduction

The last chapter of this thesis contains a summary about the presented work. Additionally, it provides an outlook about useful future work for a further improvement and refinement of the project framework. The research methodology will be observed for its applicability for this thesis. Some critical remarks will outline the limitations of the proposed approach and the used technologies.

9.2 Conclusion

This thesis introduces a number of novel techniques for the analysis of the creative process and support of the creative process of the user, based on the analysis results. The techniques were realised in a flexible framework, called the De Montfort Creativity Assistant. The whole framework is based on a mathematically sound underpinning in order to produce accurate results. The approach utilises a so-called creativity map to store and represent the logical structure of the creative process in an easy, but powerful fashion. The creativity map is a generalised transition system, where states represent stages of the artefact and transitions describe the activities of the user.

The main contribution of the presented research is a flexible description language, which allows describing patterns of creative behaviour. These behavioural patterns are applied on a creativity map in order to search for matches.

A knowledge management facility is used for the management and store of the knowledge about the analyses and the creative process itself. It caters for the freshness, completeness, consistency, precision and trustfulness of the stored entries.

The prototype implementations of the knowledge management facility and the BPDFL parser was very beneficial to determine the practicability of the presented approach. The design of the BPDFL parser enabled the development of a powerful analysis framework. It is possible to add new functionalities with only minor changes to the framework. The pluggable design allows the integration of these frameworks into the DMCA. It has not been integrated so far as it requires some major techniques, which are the topic of another thesis, written simultaneously to this thesis. This includes a framework for the creation of the creativity maps and partial creativity maps as well as the frequent behaviours. The creativity maps presented in the case studies were created by hand.

9.2.1 Review of the Research Questions

Five research questions have been stated at the beginning of this thesis. The aim of this review is to find out whether these research questions have been answered in this thesis.

1. How can information about creative behaviour, which is hidden inside the creativity map, be characterised?

The creative behaviour of a person is expressed inside the creative process, which is represented through the creativity map in the presented framework. The creative behaviour is a non-linear phenomenon. The description of creative behaviour requires a flexible model as the possible forms of these structures are unlimited. The in Chapter 4 introduced behavioural patterns enable the description of constructs (or structures), based on transitions. Behavioural Patterns are, therefore, capable of describing every type of creative behaviour. Behavioural patterns are used and applied by a parser in order to collect elements from the creativity maps, which fit to the description. Through this, it is possible to retrieve information about the creative behaviour.

2. Which computer-aided technique enables the extraction of information from creativity maps?

The information about the creative process, stored in the creativity maps, is usually of a huge size. This requires a computer-supported analysis approach. A description language for the extraction of information is introduced in this thesis. The language allows a fully flexible analysis of the creativity map. The syntax of the description language has an Extended Backus-Naur Form style notation. This enables a quick and flexible, computer-readable description of behavioural pattern. A parser was developed to cater for the extraction of information from the creativity map as described by the pattern. The parser was realised through the combination of well-known techniques such as Parse Trees, Thompson's Construction [124] and Subset Construction. It is capable of handling massive data sets. Algorithms for the identification of behavioural patterns inside the creativity map as well as for the monitoring process have been introduced in this thesis. The algorithms and functionality of special operators like the revert operator have been explained in detail. The thesis not only contains the descriptions of the concept and algorithms itself, but presents an implementation of the techniques into the knowledge management and analysis facility.

3. Which approach enables the management and store of the knowledge about creative behaviour?

The management and store of knowledge is one of the key assets in the domain of creativity analysis and support. Being able to store the information also enables the exchange of information and knowledge about creative processes. So far, there has not been a knowledge repository for the management of knowledge about creativity and creative processes. The in this thesis presented knowledge repository is a novelty. It enables the storage of the behavioural patterns, which are used for the analysis process. The facility supports the tagging of the entries with additional metainformation. It caters for the exchange of knowledge in order to improve and enrich the knowledge about the creative process and creativity itself. The implementation of the knowledge repository uses industrial-strength technologies, like a database management system (DBMS) for the storage of the pattern expressions and metainformation as well as ontologies like Wordnet for the reduction of synonymy and polysemy when comparing knowledge entries. The GUI of the knowledge repository is introduced in Chapter 7. Technical details can be found in the appendix.

4. What type of information can be extracted from the creativity maps?

A creativity map represents the complete creative process of a creator working on an artefact. The creative process is beginning with the first thoughts about the artefact and ending with the last step of completing it. The information, stored in the creativity maps, is not only consisting of creational activities which directly change the artefact, but also intellectual activities. The latter type is even more interesting as it enables the reconstruction of the creator's thoughts and the factors which influenced his or her (creative) thinking. Chapter 6 contains a detail introduction into the analysis of the creative process via the application of behavioural patterns on creativity maps. Two analysis types were introduced in that chapter: the detailed analysis and the monitoring process. The detailed analysis is used for the investigation of the complete creativity map or a partial creativity map. This type of analysis allows the application of the complete set of behavioural patterns. Furthermore, the detailed analysis also enables the comparison of two or more creativity maps, either from the same or from different creators. The monitoring process, in opposite, is only used to investigate the user's current creative process. Through this, it is possible to identify a certain behaviour at the moment it takes place. This analysis is designated for an active support of the creative process. A third analysis is focussing on frequent behaviours. These are often re-occurring structures inside the creativity map. Obviously, it is very important for the creator to find out, whether these behaviours represent poor creative behaviour or wanted creative behaviour.

5. How can the results of the analyses be used for the support of the creative process?

It was explained that the detailed analysis allows to take a deep insight into the creative process of the creator. Through this, it is possible to determine those behaviours which are beneficial for the creative process and therefore the creativity of the user. Moreover, it is also possible to find out which behaviours have a negative influence on the creative process. The flexibility of behavioural patterns and the Behavioural Pattern Description Language enables the definition of new behavioural patterns, based on the knowledge gained from previous analyses. Results of the analysis can be directly transformed into behavioural patterns, aiming to support the creative process.

The support of the creative process is realised through the monitoring of the creator's current creative process when working on an artefact. The monitoring process is basically comparing the currently undertaken activities with a set of behavioural patterns, chosen by the creator. The monitoring process will notify the creator, if a behavioural pattern, e.g. representing poor creative behaviour, is identified. In this case, the process can guide the creator to take another direction with his approach.

9.3 Review of the Research Methodology

This section discusses the research methodology for this thesis. The research methodology is important for a successful completion of a thesis. Therefore, it is useful to provide a review and discuss the encountered advantages and disadvantages. The first part of this section will discuss the adequacy of the used research methodology and the second part will discuss the reasons for selecting the case studies and their validity.

9.3.1 Adequacy of the Research Methodology

The chosen research methodology was suitable for this thesis. No major difficulties have been encountered during the whole process. The selected background material provided a good entry into the domain of creative technologies. Furthermore, it allowed identifying the weaknesses of current approaches. Through this, it was possible to take those weaknesses into account for the development process. As a result, these are avoided in the here presented approach.

The project itself required the usage of the constructive research method. It was mentioned before, that the combination of a theoretical model with a software framework, which proves the functionality of the model, was very beneficial. As a result, a number of experiments have been conducted. The results provided a very positive impression about the effectiveness of the approach.

9.3.2 Steps in Constructive Research

This section will review how each of the steps in constructive research has been covered in the thesis.

1. Select a relevant problem with enough research potential.

The research project has been carefully chosen. Research on the field of analysis and support of creativity has so far mostly consisted of philosophical discussions about the creative human and creative artefact. As a result, the study of the creative process and the therein expressed creative behaviour has been neglected, especially in the domain of computer science. This thesis has closed this research gap.

2. Conduct a thorough literature review.

The literature in Chapter 2 provides a thorough introduction into the main topics of creativity, creative processes, Creativity Support Tools and so forth. Furthermore, it contains a critical discussion of current approaches. The limitations of those approaches have been explained.

3. Construct a solution approach.

A solution for the analysis and support of the creative process has been described in Chapter 4, 5, 6 and 7. The first three chapters discuss the theoretical model of the approach. Chapter 7 contains the introduction of the tool support for the solution approach.

4. Show the theoretical connections and the research contribution of the approach.

The presented approach represents a novelty for tackling the problems of studying the nature of the creative process and creativity. It will raise awareness about the importance of the creative process, in regard to the quest for unveiling the secrets of creativity [75].

5. Examine the scope of applicability of the approach.

This step is covered by Chapter 6. Moreover, the domain-independency of the presented approach is the central theme of the thesis. The case studies in Chapter 8 show that the domain-independency has been realised successfully.

6. Demonstrate that the developed approach works.

Three case studies are presented in Chapter 8. Each of the case studies aimed to cover a different aspect, like the comparison of creative behaviours of different writer, changes in the creative behaviour of the same person when facing a different task/problem and the applicability of the approach when using secondary input material.

9.3.3 Choice of Case Studies

Chapter 9 contains three case studies, each with a different focus. The first case study contains an analysis of the creative processes of creative writers, participating in workshop. This case study provides an insight about the differences in the creative behaviours of the participants. The second case study contains the analysis of the creative processes which lead to the creation of the Chapter 2 and Chapter 8 of this thesis. The last case study contains the analysis of the creative process with data from an audio recording. The aim of this experiment is to answer the question, whether it is possible to map the creative process based on secondary material like audio recordings or notes. A fourth case study about creativity and collaboration would have been a good addition. However, there was not enough data available for such a case study at that point. It is necessary to conduct this type of analysis at a later point in future.

9.4 Critical Remarks

This section contains some information about the limitations of the presented framework. The aim of the project is to provide a balanced set of functions. Nonetheless, there are some minor limitations for some of these.

9.4.1 Data Collection Method

The current mapping of the human creative process is based on the capturing of the activities, which are undertaken during the creation of an artefact. The method for the capturing is very important for the precision of the collected data. In an ideal case, the method would guarantee the capturing of each activity undertaken by the creator. Unfortunately this is not possible in a normal working environment. The capturing method must keep a balance between the expense of collecting the data and a sufficient amount of data. The method must not interfere with the creative process itself.

There are different approaches for the capturing process. One approach was mentioned in Chapter 2: A group of researchers used cameras to record the activities of the creators. This approach is not very useful as it requires special equipment. Furthermore, it is necessary to review the whole process again in order to note the activities of the creator. The review process will take at least as long as the creative process itself.

The in this thesis presented approach will not require any additional equipment other than the in anyway used computer. The creative process is captured in real-time. This enables a monitoring process for a direct support of the creative process. It is possible that some users of the DMCM will feel distracted by the necessary click on a button. Nonetheless, the feedback from people who tested the system was very positive and no major distractions were witnessed.

It was documented in one case study, that it is possible to use records of creators for the creation of a creativity map. In the provided example, the record was an audio recording. Nonetheless, it also suggests that other record types will be useful for this task. This possibility is very valuable for a comparison of creativity in the past with that of today. Moreover, it shows the flexibility of the model that is used for the store and representation of the creative process.

9.4.2 Validity of the Analyses

At this point it is difficult to make statements about the effectiveness of the analysis processes. The here introduced framework was developed for the capturing of activities during the creative process and the extraction of information about the creative process. The initiator of the analysis himself is responsible for the selection of required information. Moreover, he must perform a precise examination of the collected information.

However, a knowledge management facility is provided which caters for the exchange of knowledge about creative processes. The analysis processes will improve with the growing amount of knowledge inside the knowledge management facility. The DMCA raises the awareness of each user for working with a collaborative environment. This will be an additional encouragement for the users to contribute their knowledge to the community.

9.5 Future Work

The development of the De Montfort Creativity Assistant is not finished with the creation of this thesis. The development of new platforms and new technologies will require the refinement of the tool-set in order to provide the best support for the user. Furthermore, it is necessary to refine the current framework and to conduct more experiments.

9.5.1 Further Data Collection

The analysis of the creative process with the DMCA is based on the mapping of the creative process. The quality of the analysis depends on the knowledge of about the creative process. The knowledge itself increases through the experience of the people of a domain. Hence, it is necessary to collect as many qualified data as possible. The benefits of the DMCA project will grow with the number of people using it and the amount of data, which was created by those people.

The De Montfort Creativity Assistant is currently used for teaching purposes in a primary school. This is a good way to generate a consistent amount of data, which is created under the supervision of a teacher. Analysing the creative process of children might be even more promising due to their still natural and uninfluenced working process.

9.5.2 Additional Capturing Techniques

It was already explained in this chapter, that the capturing techniques need to store the data in real-time in order to enable some important features for the support of the creative process. Nonetheless, it is necessary to continue looking for ways to improve the activity capturing process. Although the current approach was already proven as comfortable, there is still room for improvement in the usability for the user. For example, it might be

useful to choose an activity through hitting a shortkey combination instead of clicking a button. Through this, creators which make a heavy usage of the keyboard, like creative writers, could save time and experience fewer interruptions in their creative process.

Other approaches like eye-tracking or voice input seem not to be ready yet for this task. Once again, these approaches would require additional equipment. This seems not to be applicable for future generations of the DMCA, which might be used on mobile devices like smartphones.

9.5.3 Support of Novel Platforms

There has been an interesting development on the smartphone and tablet PC market during the last couple of months of writing this thesis. A new generation of smartphones with large screens and powerful processors allow the operation of almost any type of applications. These devices use modern operating systems with multi-tasking functionality. Tablet PCs have turned from content creating into content consuming devices. This makes them more attractive for private customers and therefore will lead to a wider availability of these devices.

Wireless broadband internet connectivity enables the users of both device types to access and share information whenever and wherever they want. This benefits new types of collaboration and user-interaction.

The De Montfort Creative Environment and especially the De Montfort Creativity Mapper need to be made available to this new type of platforms. The usage of smartphones for the mapping of the creative process will enable creators to capture activities of projects without the need of a desktop or laptop computer. Until now, the capturing process is bound to more or less stationary computers and laptops, which limits the field of application. Smartphones would help to overcome this limitation.

The flexible client-server architecture makes the DMCA is predestinated for being migrated onto different types of platforms. Moreover, it provides a central store for the data and information of the user. This allows the user to switch between different platforms as the task requires, always using the same application with access to the same resources.

References

- [1] Maryam Alavi and Dorothy E. Leidner. Knowledge management systems: issues, challenges, and benefits. *Communications of the Association for Information Systems*, 1(7):1–37, February 1999.
- [2] Robert S. Albert and Mark A. Runco. *Handbook of Creativity*, chapter A History of Research on Creativity, pages 16–31. Cambridge University Press, 2009.
- [3] Teresa M. Amabile. Effects of external evaluation on artistic creativity. *Journal of Personality and Social Psychology*, 37(2):221 – 233, 1979.
- [4] Teresa M. Amabile. Social psychology of creativity: A consensual assessment technique. *Journal of Personality and Social Psychology*, 43:997–1013, 1982.
- [5] Teresa M. Amabile. The social psychology of creativity: A componential conceptualization. *Journal of Personality and Social Psychology*, 45(2):357–376, August 1983.
- [6] Teresa M. Amabile. Social influences on creativity: Evaluation, coaction, and surveillance. *Annual Convention of the American Psychological Association*, 92:24–28, August 1984.
- [7] Teresa M. Amabile. How to kill creativity. *Harvard Business Review*, 76(5):76–87, September 1998.
- [8] Pedro Antunes and Nuno Guimaraes. A distributed model and architecture for interactive cooperation. In *proceedings of the 4th workshop on future trends of distributed computing systems*, 1993.
- [9] John Baer, James C. Kaufman, and Claudia A. Gentile. Extension of the consensual assessment technique to nonparallel creative products. *Creativity Research Journal of Creative Behavior*, 16(1):113–117, 2004.

-
- [10] Frank Barron and David M. Harrington. Creativity, intelligence, and personality. *Annual Review of Psychology*, 32:439 – 476, 1981.
- [11] Gene Bellinger, Durval Castro, and Anthony Mills. Data, information, knowledge, and wisdom, 2004.
- [12] Michael Bieber, Douglas Engelbart, Richard Furuta, Starr Roxanne Hiltz, John Noll, Jennifer Preece, Edward A. Stohr, Murray Turoff, and Bartel Van De Walle. Toward virtual community knowledge evolution. *J. Manage. Inf. Syst.*, 18(4):11–35, 2002.
- [13] Dines Bjoner. *Software Engineering 2: Specification of Systems and Languages*. Springer Berlin Heidelberg, 2006.
- [14] Margaret A. Boden. *The Creative Mind - Myths and mechanisms*. Routledge, 1990.
- [15] Chan Bok. Developing an idea processor in prolog. In *Proceedings of the 1st Visual Prolog Application & Language Conference*, 2006.
- [16] Uwe M. Borghoff and Remo Pareschi. Information technology for knowledge management. *Journal of Universal Computer Science*, 3:835–842, 1998.
- [17] W. Brauer, G. Rozenberg, and A. Salomaa. *Automata and Machines*. Texts in Theoretical Computer Science an EATCS Series. Springer Berlin Heidelberg, 2007.
- [18] Stephen Brown. Re-engineering the university. *Open Learning*, 17(3):231 – 243, 2002.
- [19] Stephen Brown. Do richer media mean better learning? a framework for evaluating learning experiences in museum web site design. *International Journal of Heritage Studies*, 12(5):412–426, September 2006.
- [20] N. Bryan-Kinns and J. G. Sheridan. Supporting mutual engagement in creative collaboration.
- [21] Nick Bryan-Kinns, Patrick G. T. Healey, and Joe Leach. Exploring mutual engagement in creative collaborations. In *C&C '07: Proceedings of the 6th ACM SIGCHI conference on Creativity & cognition*, pages 223–232, New York, NY, USA, 2007. ACM.
- [22] Keno Buss, Sascha Westendorf, and Hussein Zedan. The effect of collaboration and co-creation on the creative processes. In *Proceedings of The Second 2009 International Conference on Creativity and Innovations in Software Engineering (CISE)*, 2009.

-
- [23] Murat Cakir, Fatos Xhafa, Nan Zhou, and Gerry Stahl. Thread-based analysis of patterns of collaborative interaction in chat. In *Proceeding of the 2005 conference on Artificial Intelligence in Education*, pages 120–127, Amsterdam, The Netherlands, The Netherlands, 2005. IOS Press.
- [24] Linda Candy. Computers and creativity support: knowledge, visualisation and collaboration. *Knowledge-Based Systems*, 10(1):3 – 13, 1997. Information Technology Support for Creativity.
- [25] Linda Candy and Ernest Edmonds. Creative expertise and collaborative technology design.
- [26] Linda Candy and Ernest A. Edmonds. Supporting the creative user: a criteria-based approach to interaction design. *Design Studies*, 18(2):185 – 194, 1997.
- [27] Linda Candy and Koichi Hori. The digital muse: Hci in support of creativity: ”creativity and cognition” comes of age: towards a new discipline. *interactions*, 10(4):44–54, 2003.
- [28] E. Rodney Canfield and Guangming Xing. Compact thompson’s nfa, 2007.
- [29] Z. Chen. Toward a better understanding of idea processors. *Information and Software Technology*, 40(10):541 – 553, 1998.
- [30] Roy Yong-Joo Chua and Sheena S. Iyengar. Creativity as a matter of choice: Prior experience and task instruction as boundary conditions for the positive effect of choice on creativity. *Journal of Creative Behavior*, 42(3):164–180, 2008.
- [31] Ching-Hua Chuan. Supporting compositional creativity using automatic style-specific accompaniment. In *Proceedings of the CHI Computational Creativity Support Workshop 2009*, 2009.
- [32] Marc El-Bèze Claude de Loupy. Managing synonymy and polysemy in a document retrieval system using wordnet. In *International Conference on Language Resources and Evaluation*, May 2002.
- [33] John W. Creswell. *Research Design: Qualitative, Quantitative, and Mixed Method Approaches*. SAGE, second edition, 2003.
- [34] M Csikszentmihalyi. Implications of a systems perspective for the study of creativity. *Handbook of Creativity*, pages 313–335, 1999.
- [35] Norman K. Denzin and Yvonna S. Lincoln. *Handbook of Qualitative Research*. SAGE, third edition, 2005.

-
- [36] Pierre Dillenbourg. What do you mean by collaborative learning? In Pierre Dillenbourg, editor, *Collaborative-learning: Cognitive and Computational Approaches*, pages 1–19. Elsevier, Oxford, 1999.
- [37] Teresa M. Dobson and John Willinsky. *The Cambridge Handbook of Literacy*, chapter Digital literacy, pages 286–312. Cambridge University Press, 2009.
- [38] Paul Dourish and Victoria Bellotti. Awareness and coordination in shared workspaces. In *Proceedings of the 1992 ACM conference on Computer-supported cooperative work - CSCW '92*, pages 107–114, New York, New York, USA, 1992. ACM Press.
- [39] W. Duch. Computational creativity. In *Neural Networks, 2006. IJCNN '06. International Joint Conference on*, pages 435–442, 0-0 2006.
- [40] Ernest A. Edmonds, Alastair Weakley, Linda Candy, Mark Fell, Roger Knott, and Sandra Pauletto. The studio as laboratory: Combining creative practice and digital technology research. *International Journal of Human Computer Studies*, 63:4–5, 2005.
- [41] Douglas C. Engelbart. Toward augmenting the human intellect and boosting our collective iq. *Commun. ACM*, 38(8):30–32, 1995.
- [42] Gijsbert Erkens, Gellof Kanselaar, Maaïke Prangmsma, and Jos Jaspers. Using tools and resources in computer supported collaborative writing. In *CSCL '02: Proceedings of the Conference on Computer Support for Collaborative Learning*, pages 389–398. International Society of the Learning Sciences, 2002.
- [43] Richard W. Riley et al. e-learning. putting a world-class education at the fingertips of all children. the national educational technology plan., December 2000.
- [44] Frances Hauge Fabian. Using technology to enhance creative actions in decision making. In *Proceedings of the 35th Hawaii International Conference on System Sciences*, 2002.
- [45] Ronald A. Finke, Thomas B. Ward, and Steven M. Smith. *Creative Cognition: Theory, Research, and Application*. MIT Press, October 1992.
- [46] Gerhard Fischer. Social creativity: Turning barriers into opportunities for collaborative design. In *University of Toronto*, pages 152–161. ACM Press, 2004.
- [47] Gerhard Fischer, Elisa Giaccardi, Hal Eden, Masanori Sugimoto, and Yunwen Ye. Beyond binary choices: integrating individual and social creativity. *Int. J. Hum.-Comput. Stud.*, 63(4-5):482–512, 2005.

-
- [48] Matthew Flagg, Christopher Skeels, and James M. Rehg. Computational support for creativity via capture and access. In *Proceedings of the CHI Computational Creativity Support Workshop 2009*, 2008.
- [49] Alan Garnham and Jane Oakhill. *Thinking and Reasoning*. Blackwell Publishers Ltd, 1994.
- [50] Tim Gross and Michael Koch. *Computer-Supported Cooperative Work*. Oldenbourg Wissensch.Vlg, February 2007. German Edition.
- [51] Jonathan Grudin. Cscw: the convergence of two development contexts. In *CHI '91: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 91–97, New York, NY, USA, 1991. ACM.
- [52] Joy P. Guilford. Creativity. *American Psychologist*, 5:444–454, July 1950.
- [53] David M. Harrington. Effects of explicit instructions to 'be creative' on the psychological meaning of divergent thinking test scores. *Journal of Personality*, 43(3):434 – 454, 1975.
- [54] Beth A. Hennessey. The social psychology of creativity. In *Scandinavian Journal of Educational Research*, volume 47, 2003.
- [55] Beth A. Hennessey and Teresa M. Amabile. Reward, intrinsic motivation, and creativity. *American Psychologist*, 53(6):674–675, June 1998.
- [56] Tom Hewett, Mary Czerwinski, Michael Terry, Jay Nunamaker, Linda Candy, Bill Kules, and Sylvan Elisabeth. Creativity support tool evaluation methods and metrics. Technical report, Drexel University, 2005.
- [57] Maud Hickey. An application of amabile's consensual assessment technique for rating the creativity of children's musical compositions. *Journal of Research in Music Education*, 49(3):234, 2001.
- [58] Oliver Hoffmann, Arthur Cropley, Lemai Nguyen, and Paul Swatman. Creativity, requirements and perspectives, 2005.
- [59] Timo Honkela, Samuel Kaski, Krista Lagus, and Teuvo Kohonen. Websom - self-organizing maps of document collections. In *Neurocomputing*, pages 101–117, 1997.
- [60] John E. Hopcroft, Rajeev Motwani, and Jeffrey D. Ullman. *Introduction to Automata Theory, Languages and Computation*. Addison-Wesley, second edition, 2001.
- [61] Daniel C. Howe. Rita: Creativity support for computational literature. In *Proceedings of the CHI Computational Creativity Support Workshop 2009*. ACM, 2008.

-
- [62] Nancy Ide and Jean Véronis. Word sense disambiguation: The state of the art. *Computational Linguistics*, 24(1), 1998.
- [63] B. Joestingmeier. *Cross-cultural innovation: new thoughts, empirical research, practical reports*. Oldenbourg, second edition, 2007.
- [64] Robert Johansen. Groupware: Future direction and wild cards. *Journal of Organizational Computing*, 1:219–227, April 1991.
- [65] Vera John-Steiner. *Creative Collaboration*. Oxford University Press, July 2006.
- [66] Hilary Johnson and Lucy Carruthers. Supporting creative and reflective processes. *Int. J. Hum.-Comput. Stud.*, 64(10):998–1030, 2006.
- [67] Eero Kasanen, Kari Lukka, and Arto Siitonen. The constructive approach in management accounting research. *Journal of Management Accounting Research*, 5:243 – 264, 1993.
- [68] James C. Kaufman, Joohyun Lee, John Baer, and Soonmook Lee. Captions, consistency, creativity, and the consensual assessment technique: New evidence of reliability. *Thinking Skills and Creativity*, 2(2):96 – 106, 2007.
- [69] D.J. Ketchen and D.D. Bergh. *Mechanisms and Empirical Research*. Number 2 in Research Methodology in Strategy and Management. Elsevier JAI, 2005.
- [70] F. Keyvan, H. Borjian, M. Kasheff, and C. Fellbaum. Developing persianet: The persian wordnet. In *Proceedings of the Third Global WordNet Meeting, Jeju Island, Korea*, January 2006.
- [71] Kyung Hee Kim. Can only intelligent people be creative? a meta-analysis. *Journal of Secondary Gifted Education*, 16(2-3):57–66, 2005.
- [72] Joachim Kimmerle and Ulrike Cress. Group awareness and self-presentation in computer-supported information exchange. *International Journal of Computer-Supported Collaborative Learning*, 3(1):85–97, March 2008.
- [73] T. Kohonen. The self-organizing map. *Proceedings of the IEEE*, 78(9):1464 –1480, sep 1990.
- [74] Teuvo Kohonen. *Self-Organizing Maps*. Springer Berlin Heidelberg, 2001.
- [75] Eva Labro and Tero-Seppo Tuomela. On bringing more action into management accounting research: process considerations based on two constructive case studies. *European Accounting Review*, 12(3):409 – 442, 2003.

-
- [76] James M. Laffey, Dale R. Musser, and Linda Espinosa et al. Cscl for schools that learn. In *CSCL '02: Proceedings of the Conference on Computer Support for Collaborative Learning*, pages 111–118. International Society of the Learning Sciences, 2002.
- [77] Krista Lagus, Samuel Kaski, and Teuvo Kohonen. Mining massive document collections by the websom method. *Information Sciences*, 163(1-3):135–156, 2004.
- [78] Gina Levow. Corpus-based techniques for word sense disambiguation. Technical report, University of Maryland, Cambridge, MA, USA, 1997.
- [79] Todd I. Lubart. Creativity and cross-cultural variation. *International Journal of Psychology*, 25(1):39, 1990.
- [80] Todd I. Lubart. Models of the creative process: Past, present and future. *Creativity Research Journal*, 13(3):295–308, 2000-2001.
- [81] Todd I. Lubart. How can computers be partners in the creative process: Classification and commentary on the special issue. *International Journal of Human-Computer Studies*, 63(4-5):365 – 369, 2005. Computer support for creativity.
- [82] Neil Maiden and Suzanne Robertson. Integrating creativity into requirements processes: Experiences with an air traffic management system. *Requirements Engineering, IEEE International Conference on*, 0:105–116, 2005.
- [83] Lena Mamykina, Linda Candy, and Ernest Edmonds. Collaborative creativity. *Commun. ACM*, 45(10):96–99, 2002.
- [84] Lev Manovich. *The Language of New Media*. The MIT Press, 2001.
- [85] Melanie Marchant and Stephen C. Sesko. Science into art: A study of the creative process. In *Annual Conference of the American Educational Research Association*, 1997.
- [86] Philippe Martin, Michael Blumenstein, and Peter Deer. Toward cooperatively-built knowledge repositories. In *Proceedings of ICCS 2005, 13th Int. Conf. on Conceptual Structures (Springer, LNAI 3596)*, pages 411–424, 2005.
- [87] Robert R. McCrae. Creativity, divergent thinking, and openness to experience. *Journal of Personality and Social Psychology*, 52(6):1258–1265, June 1987.
- [88] D.E. McNabb. *Research methods in public administration and nonprofit management: quantitative and qualitative approaches*. M.E. Sharpe, 2002.

-
- [89] Sun Microsystems. Java network launching protocol & api specification (jsr-56), 2008.
- [90] Rada Mihalcea, Paul Tarau, and Elizabeth Figa. Pagerank on semantic networks, with application to word sense disambiguation. In *COLING '04: Proceedings of the 20th international conference on Computational Linguistics*, page 1126, Morristown, NJ, USA, 2004. Association for Computational Linguistics.
- [91] George A. Miller, Richard Beckwith, and Christiane Fellbaum et al. Introduction to wordnet: An on-line lexical database. Technical report, Princeton University, August 1993.
- [92] J.F. Monin and M.G. Hinchey. *Understanding formal methods*. Facit Series. Springer, 2003.
- [93] Kenneth Moorman and Ashwin Ram. A model of creative understanding. In *In Proceedings of the 12th National Conference on Artificial Intelligenc. Menlo Park*, pages 74–79. AAAI Press, 1994.
- [94] Dan Morris and Jimmy Secretan. Computational creativity support: using algorithms and machine learning to help people be more creative. In *CHI '09: Proceedings of the 27th international conference extended abstracts on Human factors in computing systems*, pages 4733–4736, New York, NY, USA, 2009. ACM.
- [95] Jeanne Nakamura and Mihaly Csikszentmihalyi. Catalytic creativity: The case of linus pauling. *American Psychologist*, 56:337–341, April 2001.
- [96] A. Newell, J. C. Shaw, and H. A. Simon. The process of creative thinking. In *Symposium on Creative Thinking*. The RAND Corporation, September 1959.
- [97] Brian D. Newman. A framework for characterizing knowledge management methods, practices, and technologies. In *Proceedings of the Third International Conference on Practical Aspects of Knowledge Management (PAKM2000)*, pages 67–73, 1999.
- [98] David M. Nichols, Michael B. Twidale, David M. Nichols, Gareth Smith, Gareth Smith, Jonathan Trevor, and Jonathan Trevor. Supporting collaborative learning during information searching. In *Proceedings of Computer Support for Collaborative Learning (CSCL 95)*, pages 367–374, 1995.
- [99] NISO. *Understanding metadata*. National Information Standards Organization, 2004. ISBN 1-880124-62-9.
- [100] R. A. Ochse. *Before the Gates of Excellence: The Determinants of Creative Genius*. Cambridge University Press, 1990.

-
- [101] Daniel E. O’Leary. Enterprise knowledge management. *Computer*, 31(3):54–61, mar 1998.
- [102] Paul B. Paulus and Huei-Chuan Yang. Idea generation in groups: A basis for creativity in organizations. *Organizational Behavior and Human Decision Processes*, 82(1):76–87, 2000.
- [103] Paul E. Plsek. Working paper: Models for the creative process. Available from: <http://www.directedcreativity.com/pages/WPModels.html>, 1997.
- [104] Tony Proctor. *Creative Problem Solving for Managers*. Routledge, 2005.
- [105] Mitchel Resnick, Brad Myers, and Kumiyo Nakakoji et al. Design principles for tools to support creative thinking. In *Proceedings of the 1st Workshop on Creativity Support Tools*, 2005.
- [106] Mel Rhodes. *An Analysis of Creativity*, volume 42. Phi Delta Kappa International, 1961.
- [107] Graeme Ritchie. Assessing creativity. In *Proceedings of the AISB Symposium on AI and Creativity in Arts and Science*, April 2001.
- [108] Graeme Ritchie. The transformational creativity hypothesis. *New Generation Computing*, 24:241–266, 2006.
- [109] Carolyn Rosé, Yi-Chia Wang, and Yue Cui et al. Analyzing collaborative learning processes automatically: Exploiting the advances of computational linguistics in cscl. *International Journal of Computer-Supported Collaborative Learning*, 3(3):237–271, November 2008.
- [110] Elisabeth Rudowicz. Creativity and culture: a two way interaction. *Scandinavian Journal of Educational Research*, 47(3):273, 2003.
- [111] Ioana Rus and Mikael Lindvall. Knowledge management in software engineering. *Software, IEEE*, 19(3):26–38, may/jun 2002.
- [112] Mark Sanderson. Word sense disambiguation and information retrieval. In *SIGIR ’94: Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 142–151, New York, NY, USA, 1994. Springer-Verlag New York, Inc.
- [113] Rob Saunders and John S. Gero. How to study artificial creativity. In *C&C ’02: Proceedings of the 4th conference on Creativity & cognition*, pages 80–87, New York, NY, USA, 2002. ACM.

-
- [114] R. Keith Sawyer. *Explaining Creativity: The Science of Human Innovation*. Oxford University Press, 2006.
- [115] Ben Shneiderman. Codex, memex, genex: The pursuit of transformational technologies. *International Journal of Human-Computer Interaction*, 10(2):87, 1998.
- [116] Ben Shneiderman. Creativity support tools - accelerating discovery and innovation. *Communications of the ACM*, 50(12), December 2007.
- [117] Ben Shneiderman, Draft July, Brad Myers, and Assoc August. Creating creativity: User interfaces for supporting innovation. *ACM Transactions on Computer-Human Interaction*, 7:114–138, 2000.
- [118] Enno Siemsen, Sridhar Balasubramanian, and Aleda V. Roth. Incentives that induce task-related effort, helping, and knowledge sharing in workgroups. *Management Science*, 53(10):1533 – 1550, 2007.
- [119] G. Stahl, T. Koschmann, and D. Suthers. *Cambridge Handbook of the Learning Sciences*, chapter Computer-supported collaborative learning: An historical perspective, pages 409–426. Cambridge University Press, 2006.
- [120] Robert J. Sternberg, Mihaly Csikszentmihalyi, and Margaret A. Boden et al. *Handbook of Creativity*, volume 12. Cambridge University Press, 2009.
- [121] Robert J. Sternberg and Linda A. O’Hara. Creativity and intelligence. *Handbook of Creativity*, pages 251–272, 1998.
- [122] R.M. Thomas. *Blending qualitative & quantitative research methods in theses and dissertations*. Corwin Press, 2003.
- [123] Sue Thomas, Chris Joseph, and Simon Mills et al. Transliteracy: Crossing divides. *First Monday*, 12(12), December 2007.
- [124] Ken Thompson. Programming techniques: Regular expression search algorithm. *Communications of the ACM*, 11(6):419–422, June 1968.
- [125] Ellis P. Torrance. *The Torrance Tests of Creative Thinking - Norms-Technical Manual Research Edition - Verbal Tests, Forms A and B - Figural Tests, Forms A and B.*, 1966.
- [126] Cathy Treadaway. Digital creativity: the impact of digital imaging technology on the creative practice of printed textile and surface pattern design. *Journal of Textile and Apparel*, 4, 2004.

-
- [127] O. Uzuner. Word sense disambiguation applied to information retrieval, May 1998.
- [128] M.L. Vasquez, C.F. Gomez, A.M. Lopez, and J. Robledo, editors. *Knowledge Management Systems Assessment: A Conceptual Framework And A Methodological Proposal*. International Association for the Management of Technology, 2001.
- [129] Graham Wallas. *The art of thought*. Harcourt, Brace and Company, 1926.
- [130] Sascha Westendorf. *PhD Thesis*. PhD thesis, Software Technology Research Laboratory (STRL), De Montfort University, 2010.
- [131] Sascha Westendorf, Keno Buss, and Hussein Zedan. Mining for behavioural knowledge and information in creative processes. In *Proceedings of The Second 2009 International Conference on Creativity and Innovations in Software Engineering (CISE)*, 2009.
- [132] Keith Williamson and Patricia Riddle. Knowledge repositories for multiple uses, 1991.
- [133] Niklaus Wirth. What can we do about the unnecessary diversity of notation for syntactic definitions? *Commun. ACM*, 20(11):822–823, 1977.
- [134] David Yarowsky. Word-sense disambiguation using statistical models of roget’s categories trained on large corpora. In *Proceedings of the 14th conference on Computational linguistics*, pages 454–460, Morristown, NJ, USA, 1992. Association for Computational Linguistics.
- [135] David Yarowsky. Unsupervised word sense disambiguation rivaling supervised methods. In *Meeting of the Association for Computational Linguistics*, pages 189–196, 1995.
- [136] Hussein Zedan, Antonio Cau, Sascha Westendorf, Keno Buss, Andrew Hugill, and Sue Thomas. Mapping human creativity. In *Proceedings of the 12th Serbian Mathematical Congress, Novi Sad*, 2008.
- [137] Yong Zhao, Carol Sue Englert, and Jing Chen et al. Tele-web: Developing a web-based literacy learning environment. Technical report, University of Michigan, June 1999.

Appendix A

Implementation

Appendix A contains the technical details for knowledge repository, the BPDFL parser and monitoring process and the networking of the DMCA and its client tools. All applications have been developed with Java, version 1.6. Additional software packages, like a DBMS, have been used where it has been necessary and appropriate.

A.1 Knowledge Repository

The implementation of the knowledge repository was realised through the usage of state-of-the-art software technology. The structure of the knowledge database will be explained in this section, followed by an introduction into the structure of the knowledge management facility (KMF). The reference implementation of the facility was realised in order to test the functionality of the in this thesis introduced concept.

A.1.1 Knowledge Database

The structure and functionality of the knowledge database has already been described. This section will describe the techniques for the storage as well as the technical layout of the knowledge database. The knowledge database is based on a relational database management system (RDBMS). A relational database is the optimal choice as a data store, because it can easily map the relations of the knowledge entries, containing of the expression of a behavioural pattern and the metainformation. The concept of the Structured Query Language (SQL) is very beneficial for this approach. It simplifies the discovery of knowledge owned by person or a domain. Furthermore, the execution speed is much higher than the speed of the approaches using text or Extensible Markup Language (XML) files.

The RDBMS for the example implementation is the free available PostgreSQL server (<http://www.postgresql.org>). This product is known for its robustness and versatility. Nonetheless, it is not a problem to use any other standard RDBMS software for this task. The SQL statements were kept simple and therefore easy to adapt to other solutions.

The following entity-relationship model (ERM) illustrates the structure and the relations between the tables of the knowledge database. The tables allow the storage of all in Chapter 5 mentioned information and knowledge entries. The style of the ERM is following the Unified Modelling Language (UML) notation.

Tablename	
PK	<u>column1</u>
FK	column2

Figure A.1: ERM Example

Figure A.1 depicts an example of table in er ERM diagram. The abbreviation PK stands for the primary key of the table. Therefore, FK stands for a foreign key within the table. Columns, containing mandatory entries, are illustrated through a bold column name. The relation between the tables is represented through arrows. The arrows also describe the cardinality of the relation.

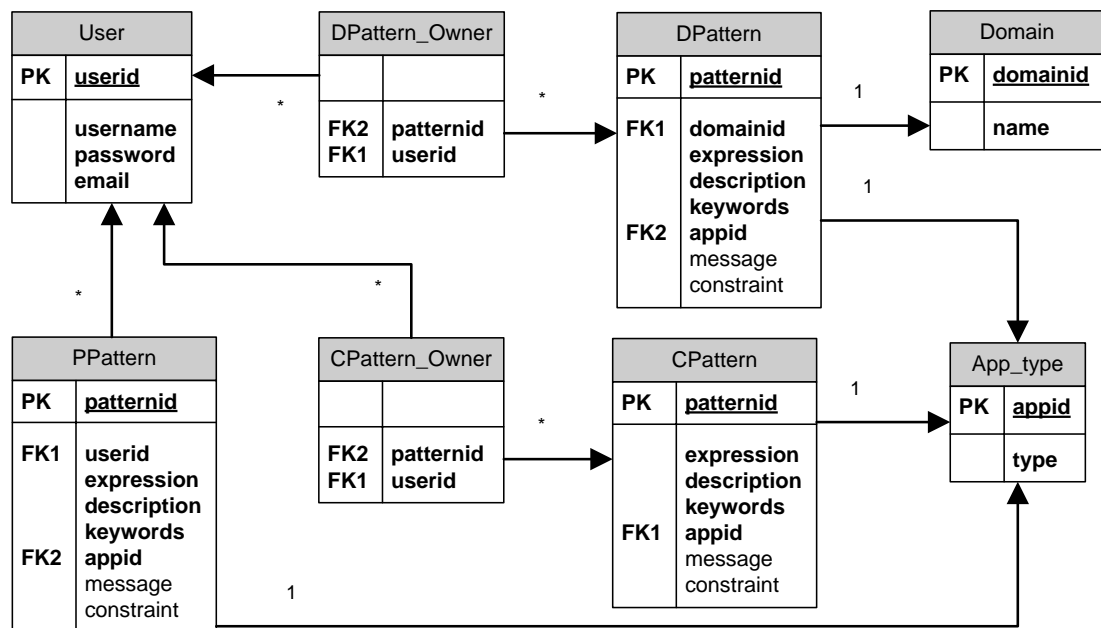


Figure A.2: ERM Diagram of the Knowledge Database

Figure A.2 depicts the ERM diagram of the knowledge database. The pattern expressions and the metainformation are stored in three different tables. The entries are stored in one of the three tables regarding their knowledge category. The table *PPattern* contains the personal knowledge entries. The tables *DPattern* and *CPattern* contain the domain and common patterns. The table *User* contains the information about the user account. The most important entry for the management is stored inside the column *userid*. This entry is used for managing the correlation of user and his knowledge entries. Interesting in respect of this are the tables *CPattern_Owner* and *DPattern_Owner*. The tables *User* and *DPattern* are connected through a m to n relation. The same as the *User* and *CPattern* tables. Through this, it is easy to identify the users of a pattern and also the pattern that are used by a specific user. The table *App_type* contains the type of creativity map the pattern expression can be applied to. An expression will be either applicable for a normal creativity map or a PCM or even both types. Table *Domain* stores the names of the domains. Through this, the creator can be provided with a list of already existing domains, which might be suitable.

The knowledge database represents only the store itself and does not contain the procedures and methods for the access and management of the knowledge. Those functionalities are only provided by the KMF.

A.1.2 Knowledge Management Facility

The following subsection will provide an overview about the structure of the KMF. The KMF caters for a modular design. The methods of the framework are defined through interfaces. The KMF is split into two parts. The first part is responsible for the user management, e.g. the log-in. The second part is responsible for the processing of knowledge itself. The facility can access the above described knowledge database through special classes.

The user management consists of one interface and two classes. One of the classes realises the interface and the second class is responsible for the access to the RDBMS. The main methods and the relations of the classes and the interface are illustrated in the following UML diagram. The items of the diagram are based on the standard UML elements. Interfaces are yellow-coloured and normal classes are blue-coloured for a better distinction.

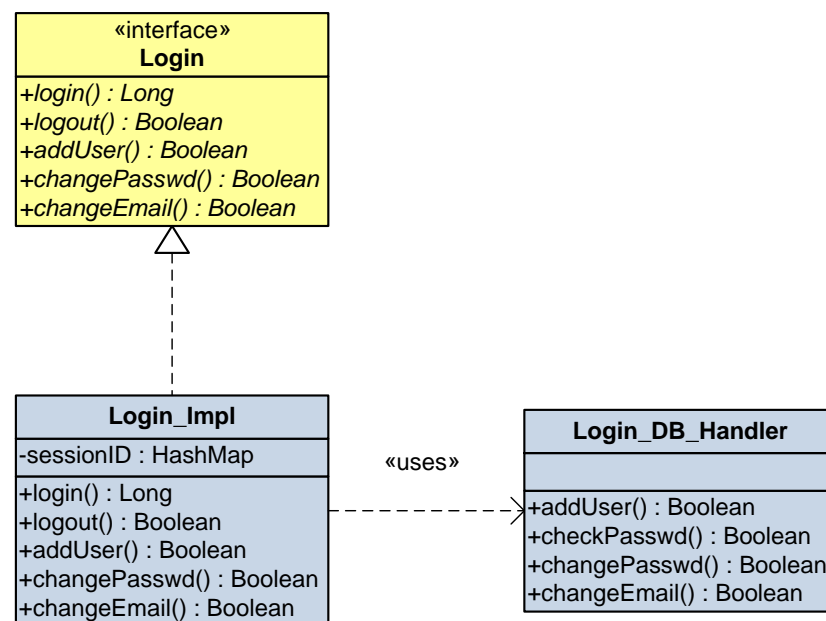


Figure A.3: UML Diagram of the Login Classes

Figure A.3 depicts the interface *Login*. The interface is realised through the class *Login_Impl*. The class *Login_DB_Handler* contains SQL statements for the retrieval of data from the RDBMS. The class *Login_Impl* acts as a mediator between the database and the client-side. Through this, the data is separated from the logic layer. The methods of the classes are self-explaining.

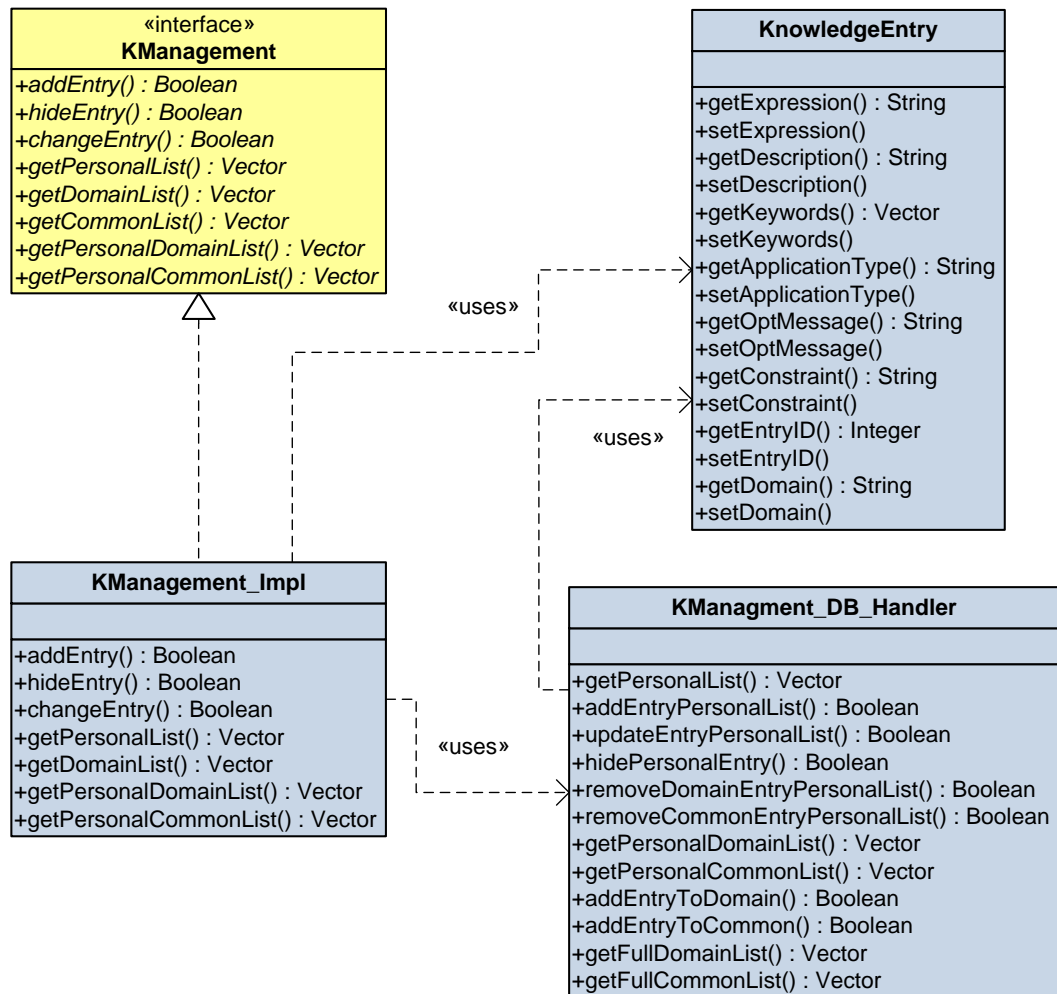


Figure A.4: UML Diagram of the Management Classes

The UML diagram in Figure A.4 depicts the relationship between the classes and the interface, which are responsible for the management of the knowledge entries. The diagram contains only public methods, as only these are necessary to explain the interaction of the classes and illustrate the functionality. The interface *KManagement* describes the methods for accessing the knowledge database. One can see that there are only a small number of public methods required in order to enable the basic operations for the knowledge management. The class *KManagement_Impl* represents the implementation of the interface. It communicates with the database handler class *KManagement_DB_Handler*.

A.1.3 Revert Storing and Retrieving

The in Chapter 7 presented identification technique enables an autonomous preparation of the revert sequences. It is not only fast and efficient, but also easy to implement. The structure of the database as well as an overview the initial implementation of the revert management is provided on the next couple of pages.

The revert sequences of a creativity map are stored in a database, so that the information will remain independent from the tracking process. A database is used, because every revert operation is stored at the moment of the occurrence. Hence, there is no identification process for retrieving all reverts of a creativity map at once. The database enables the access to the information at any time.

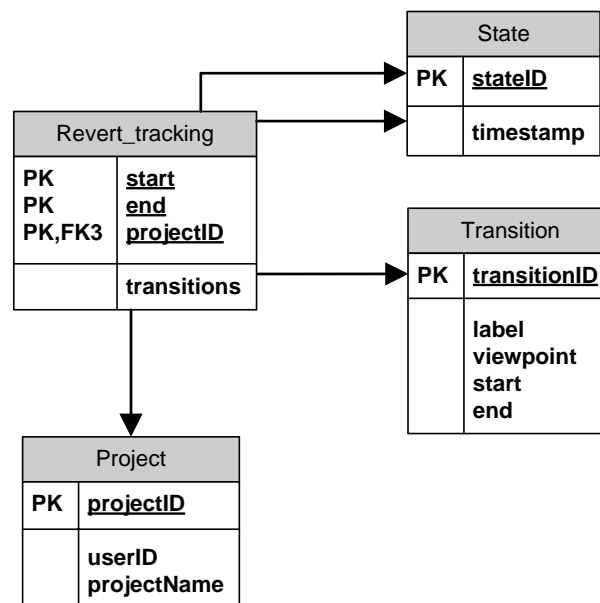


Figure A.5: ERM Diagram for the Revert Tracking Database

The ERM diagram of the database for the revert tracking process is depicted in Figure A.5. The structure of the database is very simple. Only four tables are required to store all information. The tables *State* and *Transition* contain the states and transitions, which are part of a revert sequence. The table *Project* contains the name of the project and represents a reference to the creativity map. The column *userID* is related to the table *User* of the KMF.

The following figure illustrates the architecture of the initial revert management facility.

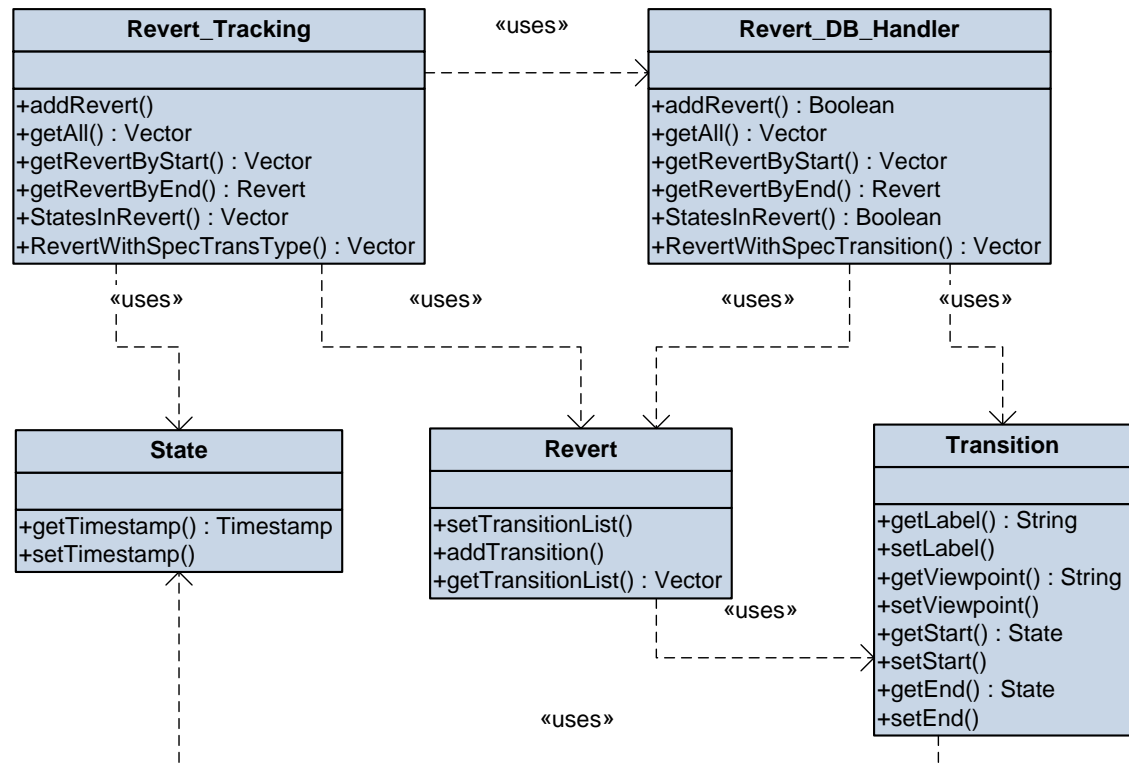


Figure A.6: UML Diagram for the Revert Tracking Facility

Figure A.6 depicts the UML diagram of the classes for the revert tracking process. The initial revert management facility consists of five classes. The main class of this facility is called *Revert_Tracking*. It provides the key functionalities of the facility. The external access to the database is provided through this class. The classes *State*, *Revert* and *Transition* are JavaBeans. These encapsulate the information. The access to the database is managed through the class *Revert_DB_Handler*. This enables a separation of the data and logic layer, as introduced before in the KMF. The revert management facility is not using any interface, as it will be only accessed through project related software components and not by external projects or facilities.

The most important and most interesting method is the `addRevert()` method of the class *Revert_Tracking*. It provides the functionalities for selecting the correct transitions and states of the revert sequence. The following source code listing contains the pseudo code for adding a new revert into the revert tracking database.

```

public void addRevert(){
    Vector start_end_states;
    Timestamp startTS = timestamp of start state;
    Timestamp endTS = timestamp of end state;
    if(first revert){
        add transitions of revert sequence to database;
        start_end_states.add(startTS);
        start_end_states.add(endTS);
    }else{
        while(iterate start_end_states){
            tmpSTS = timestamp of temporary start state;
            tmpETS = timestamp of temporary end state;
            if(tmpSTS >= startTS){
                if(tmpETS <= endTS){
                    remove sequence from current revert sequence;
                }
            }
        }
        add transitions of revert sequence to database;
        start_end_states.add(startTS);
        start_end_states.add(endTS);
    }
}

```

Listing A.1: Pseudo Code for the Identification of a Revert Sequence

The method is relatively short, despite some parts have been simplified for a better understanding. The parameters of the methods have been removed for simplicity and easier understanding. The tracking table is stored in a vector object. A vector is very beneficial for the usage in a while loop regarding the execution speed.

The code illustrates the simplicity of this revert tracking approach. The lookup and removal of previous revert sequences within the current revert sequence is realised in only one while loop.

A.2 BPDFL Parser and Monitoring Process

The BPDFL parser and the techniques for the monitoring process have been implemented in one framework. This enables a fully automatical parsing of a creativity map. The input for the parsing process consists of the pattern description in form of a BPDFL expression and the creativity map itself. The framework will parse the expression and transform it into an automaton in the way as described in Chapter 7. No external frameworks are used for the analysis process. The framework covers the in this thesis described functionality, but also provides an open architecture, which enables a further development in the future.

The functionality of this framework will finally be accessible via a GUI. This GUI will use the methods, provided by the here presented framework. The framework can be located either on the client or the server side. However, it will be most certainly located on the client, to take off the load from the server.

The following figure illustrates the relations between the classes of the framework with an UML diagram.

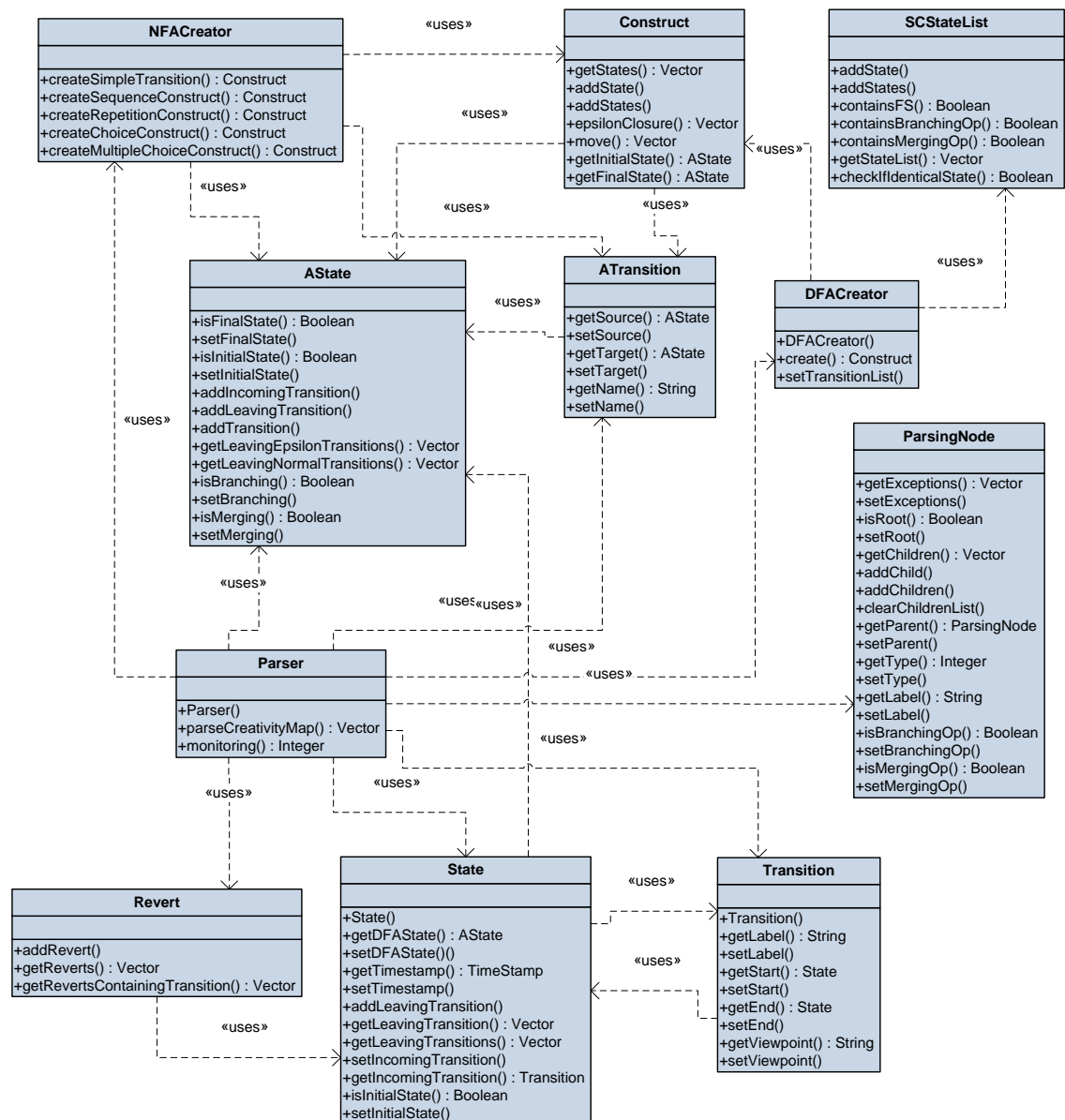


Figure A.7: UML of the Behavioural Pattern Description Language Parser

Figure A.7 depicts the UML diagram with the classes of the BPD Language Parser and the monitoring process. The class *Parser* has a central position within the framework. The methods of this class are used to initiate the parsing processes. Each instance of the class represents one behavioural pattern. The classes *State* and *Transition* are used to build the initial creativity map. Every instance of the class *State* stores the leaving transitions in a vector. This enables the storage of the creativity map in adjacency list. The classes *AState* and *ATransition* do not represent elements of the creativity map, but of the automaton. The creation of the NFA and DFA automaton is realised in the classes

NFACreator and *DFACreator*. These classes use two helper classes, namely *Construct* and *SCStateList*.

The class *ParsingNode* is used by the *Parser* class for the transformation of the the parse tree into a NFA. Finally, the class *Revert* provides the functionality for adding and accessing the information about revert sequences.

A.3 Networking of the De Montfort Creativity Assistant

A.3.1 Data Transmission over Networks

The DMCA is implemented as a client-server architecture. The tools on both client and server side are implemented as Java applications. Hence it is necessary to develop a powerful network communication platform which allows to send data from the client to the server and vice versa. Moreover, the approach must allow sending data in real-time to support the usage of collaborative applications.

The basic transmission of data between the client and the server is realised with Java Remote Method Invocation (RMI). This technology enables a reliable and stable data transmission from a client to the server. Unfortunately, the drawback of this approach is the fact, that it is not possible to call methods on both sides. RMI does not support calling methods on the client side from the server through a firewall or a network address translation (NAT) router. Unfortunately, this is often necessary. Instead, the DMCA uses a polling concept to inform the client applications about updates.

Other approaches like the new Session Traversal Utilities for NAT (STUN) technique, which could enable the call of a method on the client side, cause more disadvantages and did not reach a stable implementation to this date.

All network activities of the DMCA are treated as an event. A change or an update inside an application will trigger an event. This event is send via the network to the server. The server can also create events and store them in a queue until the client polls and retrieves the latest events.

The DMCA uses a wrapper class, which encapsulates the data as it is send over the network. The implementation of the DMCA contains the abstract class *Event*. Each application of the DMCA uses customised event classes, which extend from this abstract class, for the communication. The elements, responsible for the transmission of the data

only need to know the class *Event* itself. Hence, it is possible to add new applications to the DMCA without changing the whole implementation of the network facilities. The new application only requires modified event classes, which extend from the abstract class.

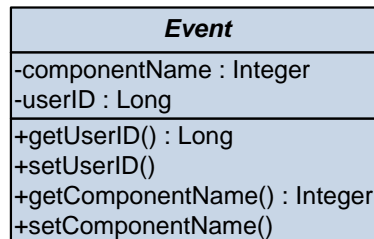


Figure A.8: Abstract Class Event

Figure A.8 depicts the UML diagram for the class *Event*. It contains only two variables and the associated getter and setter methods. These two variables are the component name and the user id. The component name is required in order to delegate the event to the correct class for the further processing on the target system. The user id is required to assign an event or action to the user who triggered the event.

A.3.2 Java Web Start

Web Start is a Java technology that allows launching standard Java applications from a website by clicking on a hyperlink [89].

In regard to normal Java applets, Web Start offers several advantages regarding to the design, implementation and runtime behaviour. Applications, initiated by Web Start are running outside the browser. This avoids a lot of restrictions regarding the usage of resources. The technology requires no additional software or changes to the browser settings on the client machine. It is only necessary to enable the Web Start support on the web-server. The hyperlink for the application is same as hyperlink to a normal website.

```
<a href="http://dmu-ca.ioct.dmu.ac.uk/dmca/ce.jnlp">...</a>
```

One can see that only the name of the file, the hyperlink is pointing to, is not a HTML file, but a description file for the Java application.

The arguments and settings for the initialisation of the Java application via Web Start are defined through the Java Network Launching Protocol (JNLP). The protocol is used to describe the resources and settings, which are required to run the application. The settings are stored in a XML structure.

The JNLP description file contains two important sections, which declare the resources and settings of the application.

```
<resources>
  <j2se version="1.6+"
    href="http://java.sun.com/products/autodl/j2se"
    max-heap-size="512m"
    java-vm-args="-Djava.rmi.server.useLocalHostname=true
      -mx256m"/>
  <jar href="ce.jar" main="true" download="eager"/>
  <jar href="lib/jdom.jar"/>
  ...
</resources>
```

Listing A.2: Resources Section of the JNLP file

The code in Listing A.2 depicts the specification of resources. This specification includes settings for the Java Virtual Machine (JVM) like the required Java version or settings for the size of the heap. Furthermore, the section contains also information about the used Java Archive (JAR) files, with at least the main application.

```
<application-desc main-class="ca.ClientMain">
  <argument>dmu-ca.io.ct.dmu.ac.uk</argument>
</application-desc>
```

Listing A.3: Application Description Section of the JNLP file

Listing A.3 depicts another important section of the JNLP file. It contains the application description, where the main class of the application is defined. It is also possible to specify arguments for the main class.

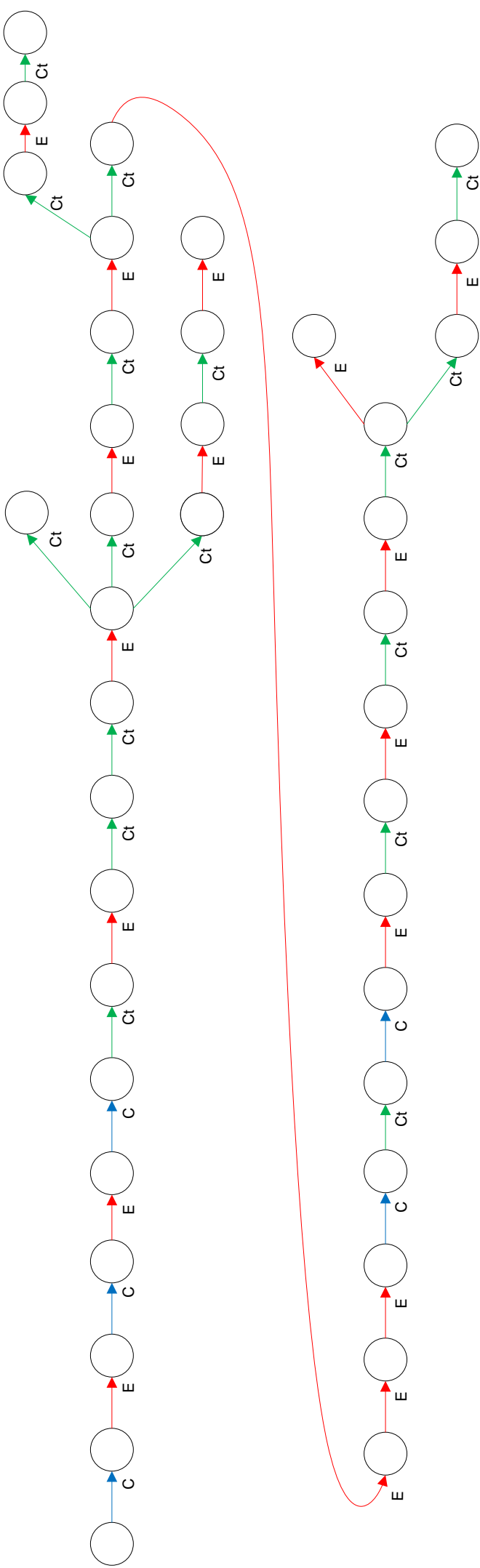
It is obvious that the application itself needs to be stored in one or more JAR container. This will also reduce the size of the package.

Java Web Start eases the distribution of updates for the application. It will automatically check at each launch of the application, if a newer version is available. Packages are automatically downloaded and updated, in case of a newer version. As a result, the user always works with the most recent version.

Appendix B

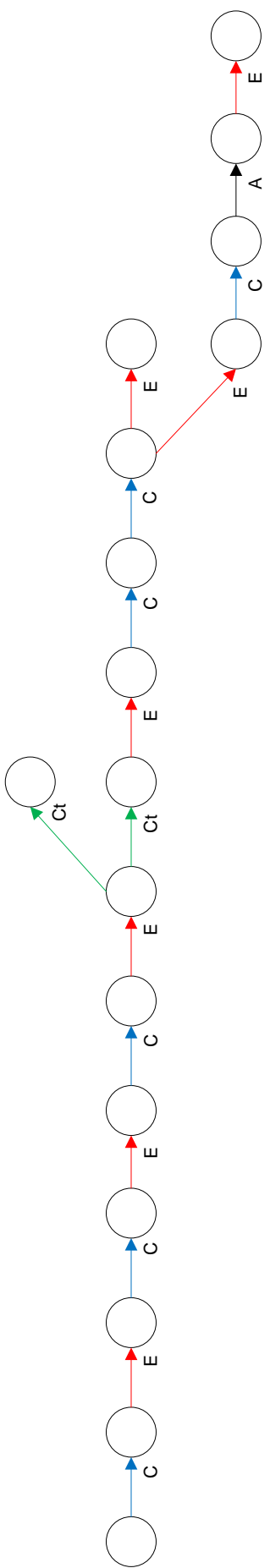
Creativity Maps of Creative Writers

The following pages depict the creativity maps of creative writers, participating in a workshop. All creativity maps represent the complete data set created during the workshop. The creativity maps are displayed in no specific order.



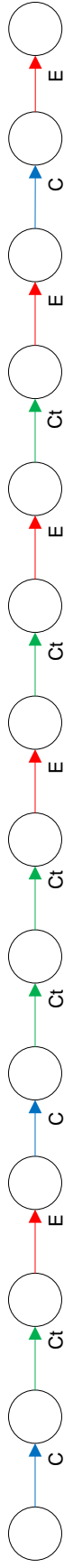
Action	Description
→ E	Editing
→ Ct	Contemplating
→ C	Conceptualising




Creative Writing Workshop – Creativity Map 1

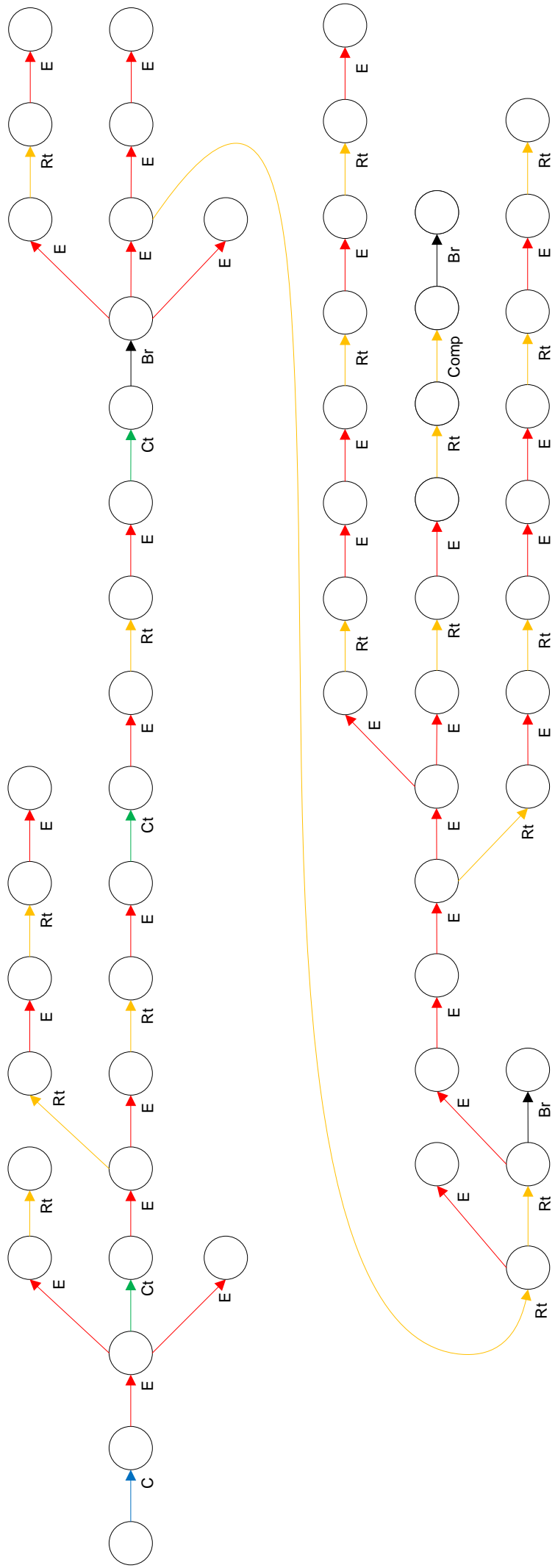


Action	Description
	Editing
	Contemplating
	Conceptualising
	Awaiting

Creative Writing Workshop – Creativity Map 2

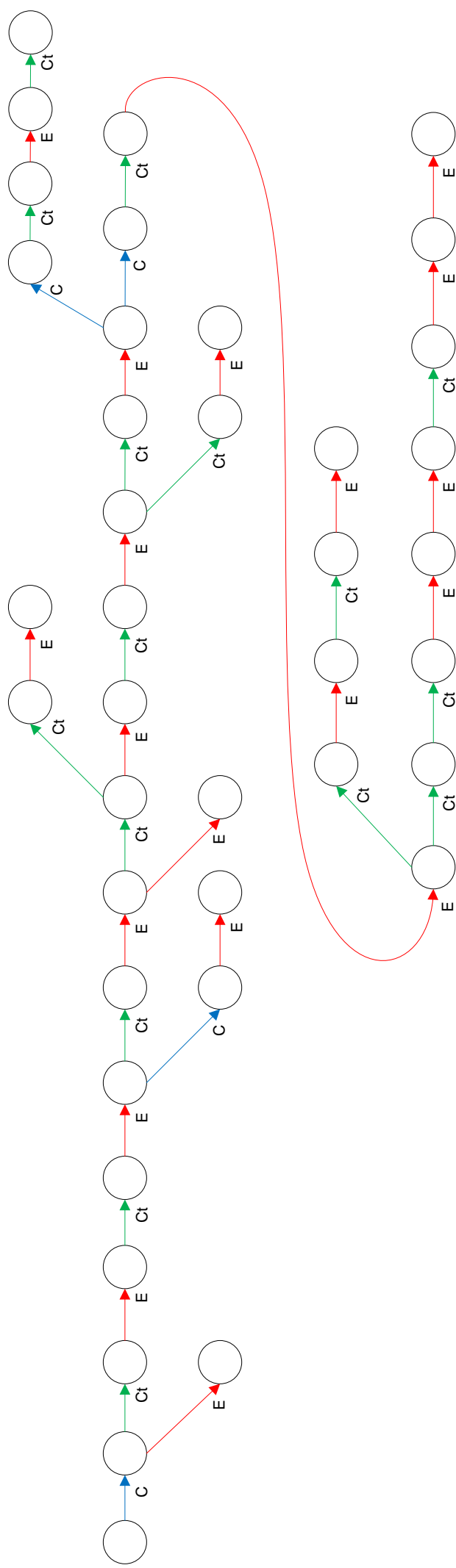





Action	Description
 E	Editing
 Ct	Contemplating
 C	Conceptualising



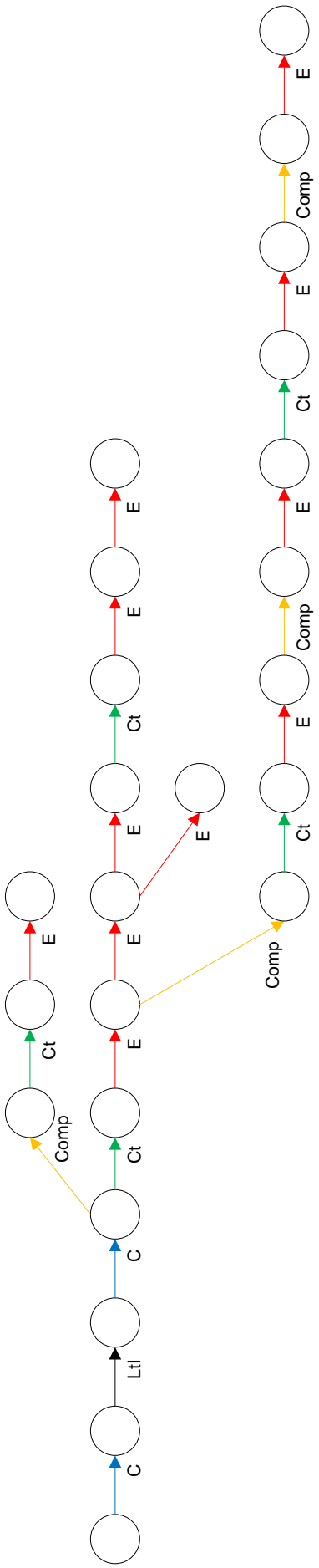
Action	Description
	Editing
	Contemplating
	Conceptualising
	Comparing
	Reading through
	Break

Creative Writing Workshop – Creativity Map 4

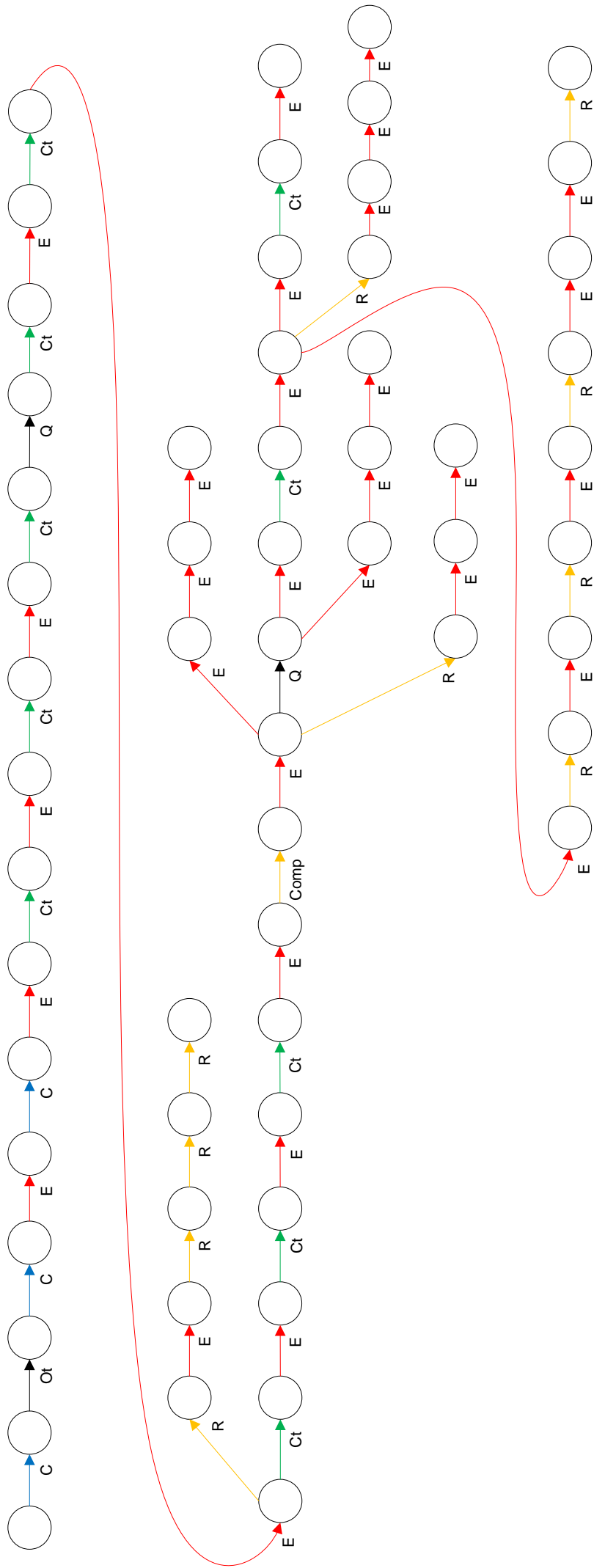


Action	Description
	Editing
	Contemplating
	Conceptualising

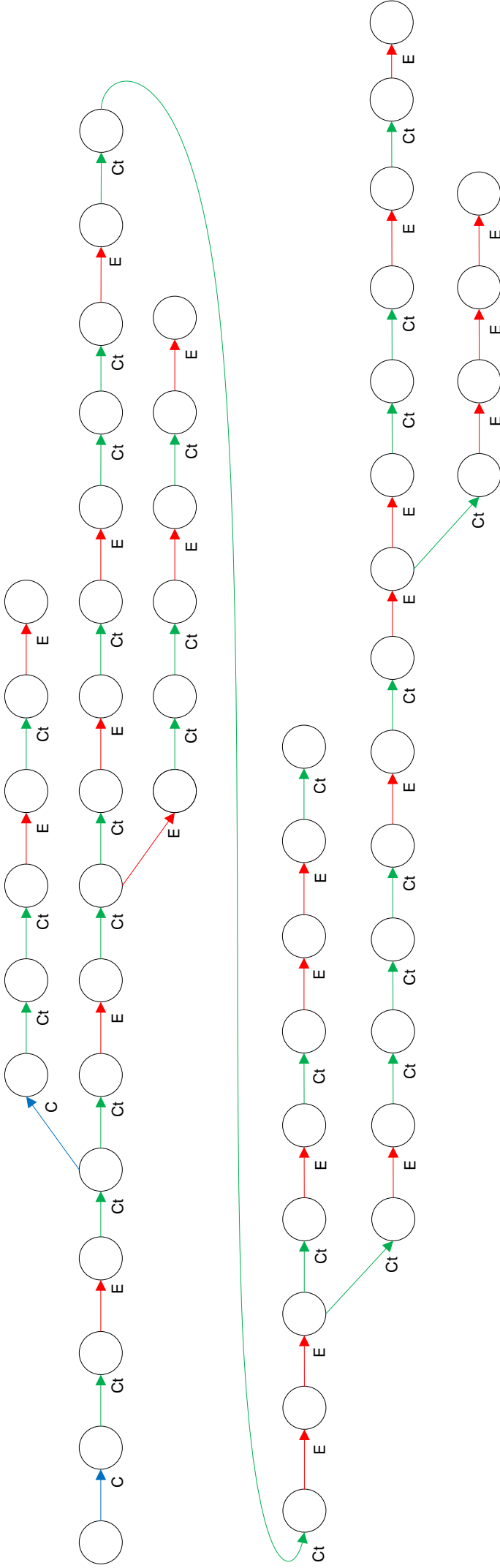
Creative Writing Workshop – Creativity Map 5



Action	Description
→ E	Editing
→ Ct	Contemplating
→ C	Conceptualising
→ Comp	Comparing
→ Ltl	Listening to Instructions

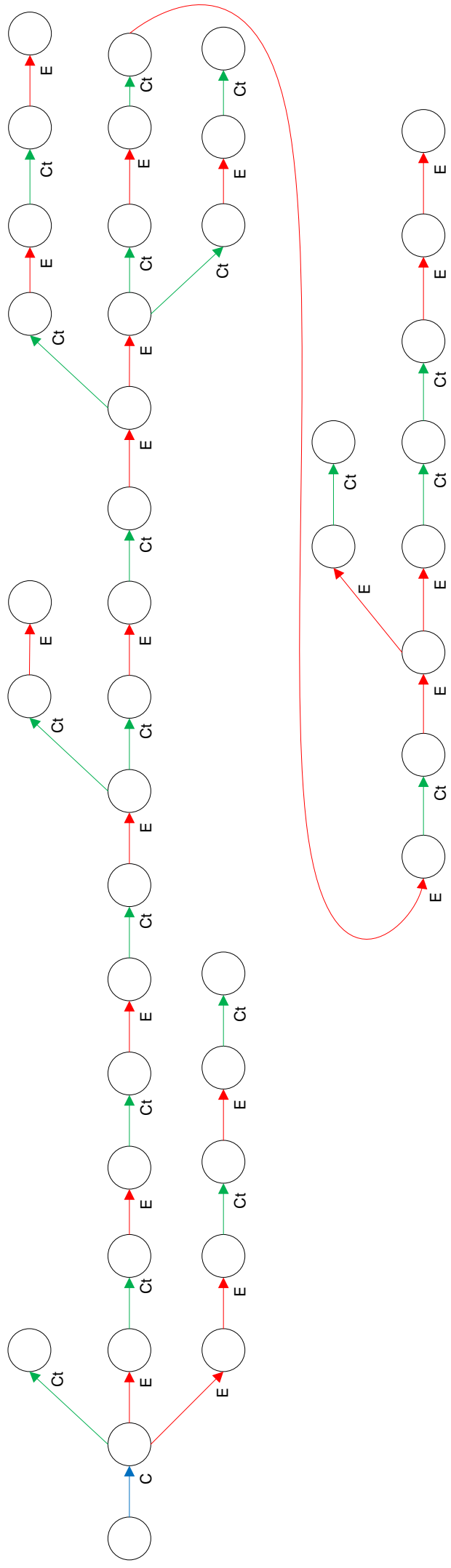


Action	Description
	Editing
	Contemplating
	Conceptualising
	Comparing
	Reading
	Questioning
	Other



Action	Description
	Editing
	Contemplating
	Conceptualising

Creative Writing Workshop – Creativity Map 8



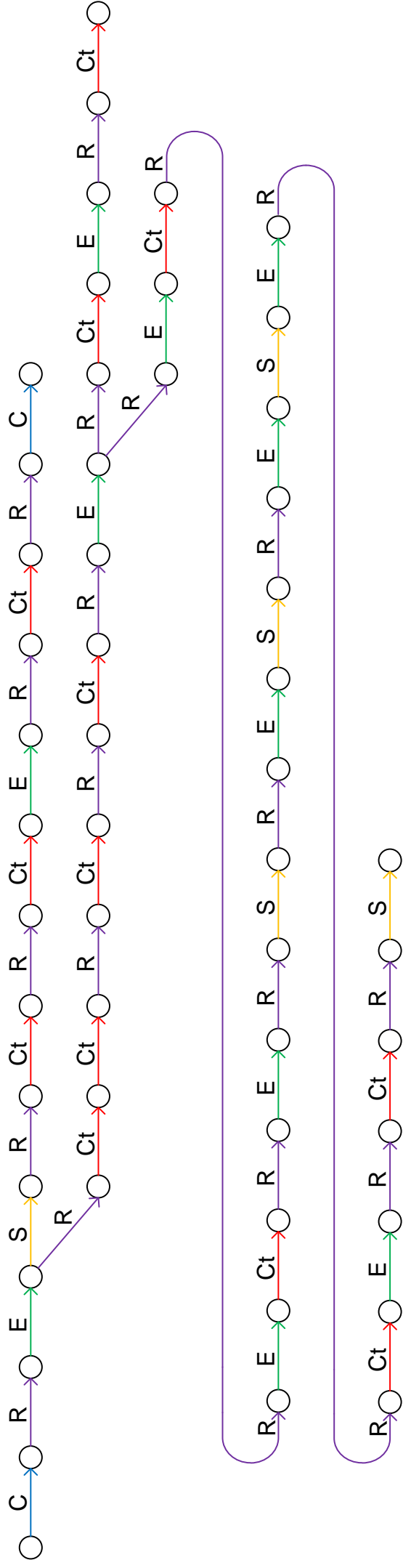
Action	Description
	Editing
	Contemplating
	Conceptualising

Creative Writing Workshop – Creativity Map 9

Appendix C

Creativity Maps for Parts of the Thesis

The following two pages contain the creativity maps for the second case study of this thesis.



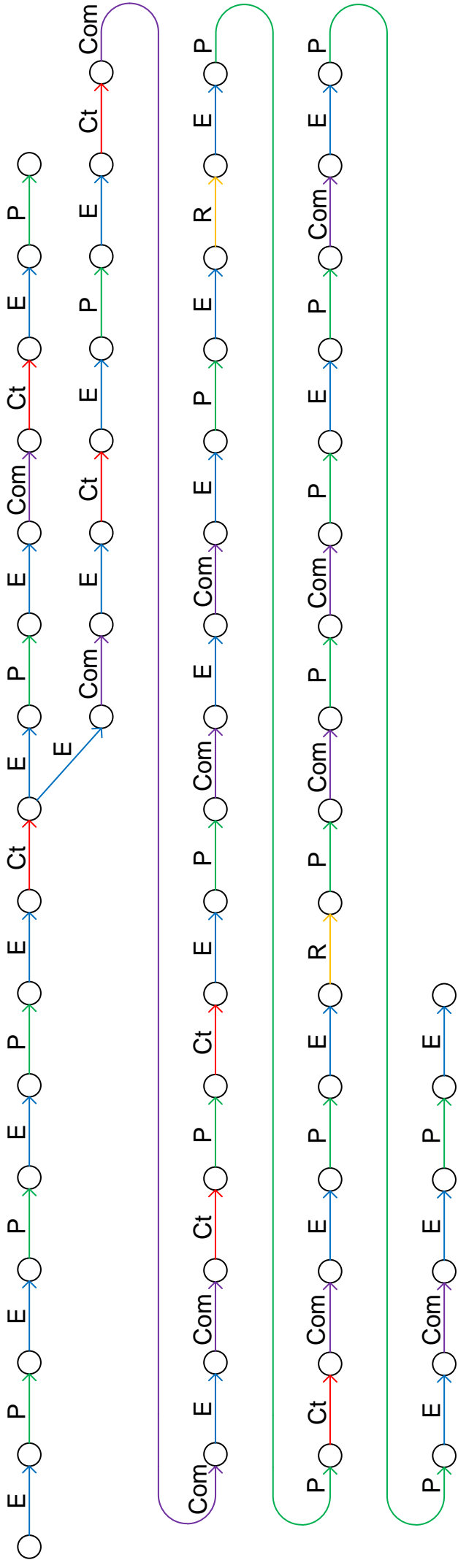
Action	Description
E →	Editing
Ct →	Contemplating
C →	Conceptualising
S →	Searching
R →	Reading

Creativity Map – Chapter 2

Appendix D

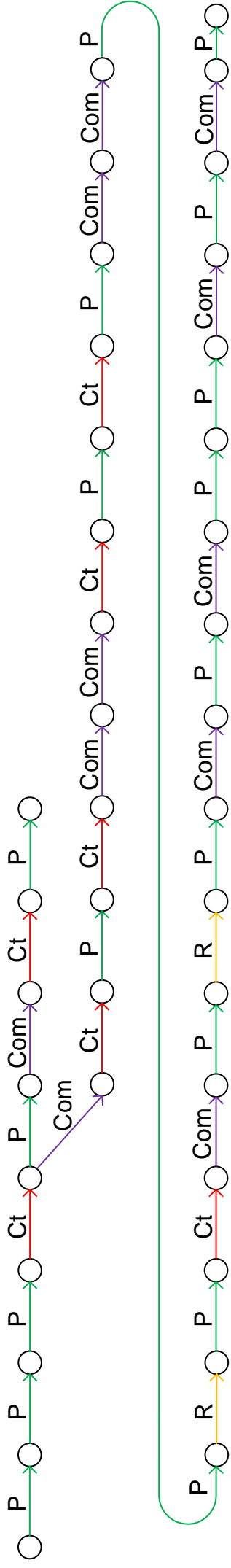
Creativity Map and Partial Creativity Map of the Musician

The following two pages contain the creativity map and partial creativity map for the third case study of the thesis.



Action	Description
E →	Explaining
Ct →	Contemplating
P →	Playing
Com →	Composing
R →	Reading

Creativity Map of the Musician



Action	Description
Ct →	Contemplating
P →	Playing
Com →	Composing
R →	Reading

Partial Creativity Map of the Musician