TOVE GUSTAVI AND MAGNUS JÄNDEL

Computational Creativity: Novel Technologies for Creative Decision Making

An introduction and literature review

FOI-R--3664--SE
ISSN 1650-1942 April 2013

Tove Gustavi and Magnus Jändel

Computational Creativity: Novel Technologies for Creative Decision Making

An introduction and literature review

Titel	Maskinkreativitet - ny teknik för kreativt beslutsfattande
Title	Computational Creativity: Novel Technologies for Creative Decision Making
Rapportnr/Report no	FOI-R—3664--SE
Månad/Month	April
Utgivningsår/Year	2013
Antal sidor/Pages	64 p
ISSN	1650-1942
Kund/Customer	FMV
FoT område	Ledning och MSI
Projektnr/Project no	E34205
Godkänd av/Approved by	Lars Höstbeck
Ansvarig avdelning	Informations- och areosystem

Sammanfattning

Kreativt beslutsfattande är ofta nyckeln till framgång men ledare är många gånger stressade och har då svårt att komma på den förlösande idén. Det är mycket lättare att utvärdera en idé när den väl ligger på bordet. Hur får vi fram nya och relevanta idéer i strategiskt och operativt beslutsfattande? Traditionella lösningar bygger på att stimulera mänsklig kreativitet med innovationskonsulter, brainstorming och kreativitetsprocesser men detta passar ibland inte för de höga kraven och tempot i militärt beslutsfattande. Kan maskinintelligens bidra med kreativa idéer? Det är nyckelfrågan i det nya forskningsfältet maskinkreativitet.

Principen för beslutstöd med maskinkreativitet är att artificiell intelligens i form av ett datorprogram stödjer en beslutsprocess genom att föreslå nya idéer. Dessa idéer kan i operativt beslutsfattande introducera nya handlingsalternativ, identifiera anomalier i beslutsfattares situationsuppfattning eller föreslå aspekter av situationen som beslutsfattare borde titta närmare på. Strategiskt beslutfattande kan till exempel stödjas med idéer till nya scenarier eller förslag till nya tekniska lösningar.

Rapporten är skriven på uppdrag av FMV och beskriver därför forskningen inom maskinkreativitet med fokus på vad som är användbart för beslutsstöd och för militärt relevanta autonoma system. I kapitel 1 förklarar vi för läsare som inte är datorvetare hur det är möjligt för maskiner att hjälpa människor med kreativa idéer och hur långt forskningen i stora drag har kommit. Vi ger exempel på hur maskinkreativitet idag används för strategisk scenarioplanering och hur operativa beslut kan stödjas.

Kapitel 2-4 är en litteraturstudie där vi beskriver viktiga metoder såsom begreppsanalogier, informationsteori, logikmetoder och genetisk programmering. Vi analyserar även vad vi kan lära oss från tillämpning av maskinkreativitet inom berättelseskapande, konst och teknikinnovation. I kapitel 5 gör vi ett försök till prognos över teknikutveckling och tillämpning inom beslutstöd och autonoma system.

Nyckelord: Beslutsstöd, autonomi, artificiell intelligens, maskinkreativitet

Summary

Creativity is often the key to success in decision making but stressed leaders find it difficult to come up with novel ideas. It is much easier to evaluate an idea once it is on the table. How do we develop new and relevant ideas in strategic and operational decision-making? Traditional solutions are based on inspiring human creativity with consultants, brainstorming and innovation processes but these approaches are not always adequate for fast-paced military decision making. Can artificial intelligence contribute creative ideas? This is what the emerging science of Computational Creativity is exploring.

Computational creativity software supports operational decision processes by proposing novel and relevant ideas including possible courses of action, anomalies in the current situation analysis or aspects of the situation that should be considered by the staff. Strategic decision making is aided with new scenarios or technical solutions.

This report is commissioned by the Swedish Defence Materiel Administration. We review computational creativity from the perspective of decision support and autonomous systems. The first chapter explains in layman's terms how it is possible for machines to support humans with creative ideas and how far the research has advanced. Examples on how computational creativity today is used for strategic scenario planning are provided and we discuss how operational decisions can be supported in the future.

Chapters 2-4 review the scientific literature on computational creativity. We analyse the tools of computational creativity including conceptual analogies, information theory, logic and genetic programming. We also examine applications of computational creativity in decision making, story generation, art, technology, and innovation. In chapter five, we forecast the evolution of technologies and applications for decision support and autonomous systems.

Keywords: Decision Support, Autonomy, Artificial Intelligence, Computational Creativity

Contents

1 Introduction

1.1 The objectives of this report

The reader of this report will in the *Introduction* get a tutorial overview intended for non-specialists. In this part we focus on explaining what computational creativity is and what it can be used for. Read only the Introduction if you are pressed for time or want to avoid technical details.

Chapter 2, *The Research Field of Computational Creativity,* describes the history and current state of computational creativity as a science. This information is useful for readers who want to go beyond this report and find out more about the field from other sources. We discuss definitions of creativity and outline relations and demarcations to neighbouring and subsuming research fields.

The next two chapters *Applications* and *Methodologies* are a literature study where we focus on the recent research that is most relevant for decision support and autonomy. By decisions, we mean all kinds of choices including technical design, acquisitions, management, operational and strategic decisions. We try to explain the gist of the research as simply and succinctly as possible but some parts of these chapters require a basic understanding of computer science. Computational creativity is a young field of research and little work has directly targeted decision support. For this study, we have therefore selected many papers in other fields of applications that demonstrate techniques that potentially could be applied to decision support. We have found that it is in some cases easier to explain the results from the point of view of the application domain while, in other cases it makes more sense to explain the methodology followed by application examples from various domains. Both chapter three *Applications* and chapter four *Methodologies* aim at explaining the state of the art in computational creativity each from the perspective that gives the reader the easiest approach to the results.

In the final chapter *Conclusions,* the authors of this report provide some closing thoughts on the present state, future prospects and potential for applications of computational creativity.

1.2 Computational Creativity for Decision Support and Autonomy

From a practical point of view, creativity is the ability to come up with novel ideas that are helpful for what we are trying to achieve. Very few real-life decisions can be made by strictly applying some formal method or logical analysis. Before we can even start employing our analytical, management, engineering or operational techniques we need the crucial idea on what kind of situation we are in and what kind of actions we can take. When we have found a good conceptual idea to guide our further analysis, creativity is needed in every step of breaking down the strategic concept to practical actions. Even as we have found a satisfactory solution it is always possible to come up with a new idea for a better and often radically improved result.

The importance of creativity is particularly evident for intended victims of strategic deception. Inventing a deception stratagem is an act of creativity and uncovering deception requires even more creativity. In 1940, German intelligence sent agents to Britain for spying and sabotaging while deceptively appearing to be legitimate refugees, often of Jewish ethnicity. British intelligence cracked the cover of most of these agents and used them for feeding false information to the Germans in sophisticated deception operations that was instrumental for the success of the D-day invasion.

Normally, we rely on gifted individuals or teams for coming up with creative ideas. Human creativity is sometimes aided by dedicated creativity methods such as brainstorming or software intended for organizing and facilitating human creativity. Idea generation is often time-consuming and expensive. Human creativity is brilliant and the pride of our species but it is also unpredictable and unreliable. Even Noble prize winning writers suffer from writer’s block that can go on for months and years. Stress, fear and exhaustion kill creativity with consequences that could be very serious in military and crisis management situations. This is the main reason why deception nearly always is successful in situations where the intended victim makes key decisions under severe stress.

In our daily life, we are supported by more and more intelligent machines. Can we use machines for supporting decision-makers with creative ideas? Some people find the concept of creative machines outrageous since creativity seems to be so close to the very core of being human. On the other hand, we could argue that machines are immune to stress, fear and exhaustion and that even highly stressed decision makers can be quite good at evaluating the merits of an idea once it is on the table. Even at their best, humans are slow and unreliable as idea generators. The combination of creative artificial intelligence and human judgement is a far-off and astounding thought but could be worth looking at. In this spirit, we will explore decision-support and autonomy applications of the burgeoning research field of computational creativity.

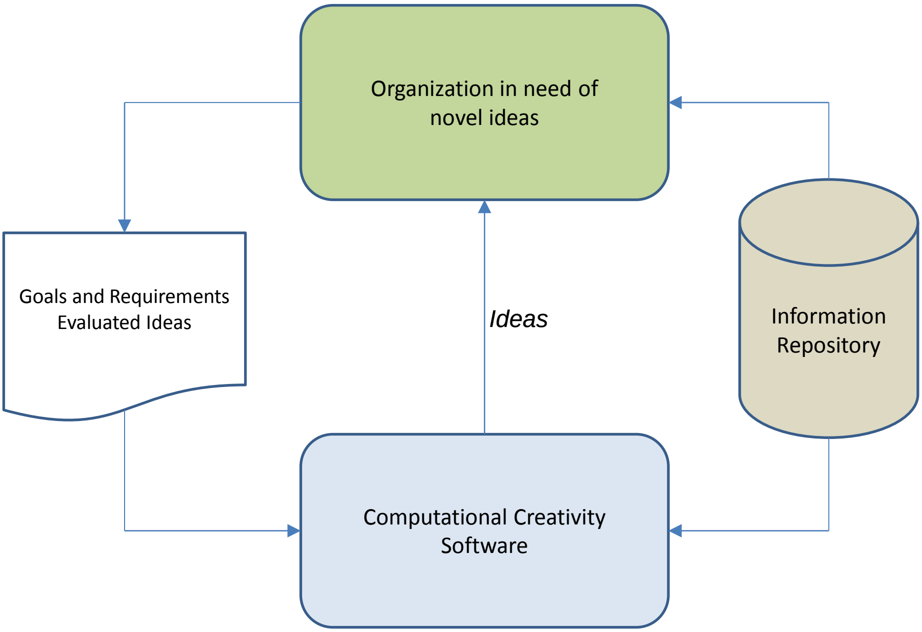

Figure 1. The concept of computational creativity. An organization gets ideas from the computational creativity software and evaluates the ideas. Goals and requirements together with feedback on previous ideas guide the software. An information repository provides both humans and the computational creativity software with background and context knowledge.

The concept of creativity is very hard to define precisely and this difficulty carries over to the computational creativity where researchers find it hard to agree on the scope and boundaries of the field. In sections 2.1 and 2.2 we provide operational definitions of creativity and computational creativity that boils down to that it is the users of the output that decide if the ideas are creative or not. Fig. 1 illustrates how creative software provides ideas to an organization.

In this report we focus on decision support which seems to be the security-related application of computational creativity that is closest at hand. Autonomous computational creativity might, however, be needed in situations where humans are not able to intervene either because a mobile robot is cut off from communication or because the pace of events is too fast for human cognition. The former situation might for example involve an unmanned military aircraft that is under attack by smart weapons and unable to communicate because of adversary electronic warfare measures. Handling a surprise attack under cover of deception requires a fair measure of creativity that in this case must be innate in the autonomous system. The latter case might for example involve autonomous software that defends against cyber-attacks. Such attacks are often very ingenious and it takes creativity to respond adequately. The attack could be over in microseconds so it is impossible for human decision makers to be in the loop for immediate defensive action. Many of the technologies that we consider in this report are relevant both for decision-support and for autonomous computational creativity.

1.3 A Success Story

To give a deeper understanding of how computational creativity can enhance human decision-making faculties we present a pioneering and successful full-scale application in some detail. Chess is the first field of human endeavour where computational creativity has had a large impact on the methods and performance of human experts. Chess is also a highly competitive sport focusing on tactical and strategic decision-making and is a game that for thousands of years has been considered to be a model of war. It is therefore worthwhile to learn from this forerunner of computational creativity in decision support.

Since the IBM supercomputer Deep Blue vanquished the world champion Gary Kasparov 1997, computer chess has evolved rapidly. A commercial off-the-shelf chess program running on a laptop is as strong as the best human players. A cell phone won a grand master tournament 2009 (the world´s top few thousand players are grand masters). It is indisputable that chess programs are strong players but it is more relevant for computational creativity that human champions use chess programs for learning how to play more creatively. Quoting world champion Vishy Anand, "*I would say nowadays it is impossible to work without computers. … It allows you to do incredibly creative things*" (Bushinsky, 2009). Computer chess expert Shay Businsky says: "*Chess played by today´s machines looks extraordinary, full of imagination and creativity*" (Bushinsky, 2009). How can a mechanical thing teach humans creativity? To understand this, we will first investigate how chess programs work and then how humans cope with the stress and complexity of grand master chess.

Chess is a board game for two players that starts from a fixed position and in which players take turns making moves according to fixed rules. The goal of the game is to capture the King of the opponent. Computers play chess by first searching in the space of possible moves and then evaluating the resulting position. The search process involves listing the computer player´s legal moves, for each such move listing the opponents possible moves, for each opponent move listing the own possible moves and so on. The result of the search is a tree with positions as nodes and moves as edges (see Fig. 2).

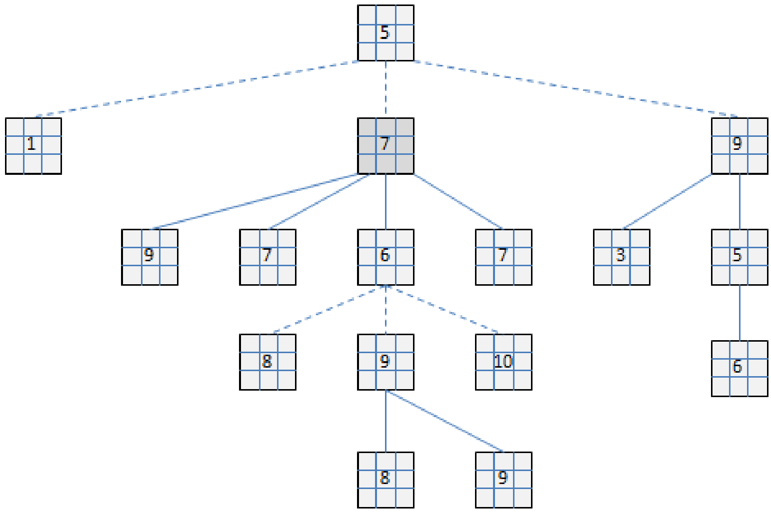

Figure 2. The search tree of a chess program. Each node symbolises a chess position. The position at the top of the figure is the present position and the program ponders the next move. The number in the centre of each position symbol is the value of the position from the program´s point of view. A higher number means that the program is more likely to win the game. The second row from the top indicates possible positions after different alternative first moves by the program. The third row shows positions after different alternative responses by the opponent. The fourth and fifth rows indicate possible positions after the next round of moves by the program and the opponent respectively. Dashed and solid lines indicate the program´s and the opponent´s moves, respectively. Positions are evaluated by the program and the search is aborted in very low-valued positions. The preferred move, showed in dark grey, is the one that leads to the position with the highest score given that the opponent selects the most favourable move from its own point of view. A real chess search tree is deeper and much more densely branched.

The game is so complex that it is typically impossible to expand the resulting tree of moves to a state where one of the players has won. Each position in the search tree is rather evaluated by a heuristic algorithm resulting in a score indicating how favourable the position is deemed to be. The search is aborted if the position scores sufficiently low, resulting in a search tree where some branches are truncated early and more promising branches are traced for a long series of moves. The chess program will eventually select a move that leads towards the best position that can be achieved even if the opponent plays optimally.

Human chess masters play partly by the same search and evaluate method as computers but also by making plans guided by strategic rules and doctrines. The human ability to search deeply in a strongly branching tree of possible futures is much more limited than that of computers, although humans still outshine algorithms in evaluating a position. To compensate for the lack of brute analytic power, novice chess players learn from historical experience embodied in doctrines and strategic principles. The “Old Russian School” was, in the years before the Deep Blue versus Kasparov competition, the pinnacle of accumulated dogmatism. A key ingredient in powerful human play is to make a good principled plan and update or reconsider the plan in critical phases of the game.

As an example of a strategic rule in chess consider the “Save the Queen rule”. The Queen is the most powerful chess piece and the rule states that the Queen should be kept safe behind a protective screen of less valuable pieces in the beginning of the game. Breaking the rule was in the pre-computer days considered to be a hallmark of incompetence. Computer programs showed that breaking the “Save the Queen rule” can be advantageous in specific positions and human players are now following suit. Gaining an advantage by breaking the stigma of conventional rules is widely taken to be a signature of creativity. In the case of the “Save the Queen rule” computers unleashed more creative play by human players.

The main reason why computers teach humans creative play in chess is that computers play more concretely. Chess algorithms do not make plans and are unaware of chess doctrines, thus looking only to the risks and opportunities of the actual present position. It is quite feasible to program computers to make plans and adhere to Old Russian School doctrines but it has been found that this weakens game performance. By ignoring cultural bias and doctrine, computers can sometimes find solutions that teach humans novel and valid approaches to playing chess. From a human point of view, this is highly valuable creative input. Since chess is very competitive, humans have not been slow to take advantage of the new tool for enhancing creativity in the human style of playing that still largely rests on plans and doctrines.

The official rules of chess competitions do not allow the direct use of computers as creative decision support. It would, however, be quite feasible to build a version of a chess program that serves as a creative assistant to a human that is playing a game of chess. Figure 1 outlines how the program could work. The program and the human would analyse the same position and the human would input the tentatively selected move in the computer. The chess program blocks the branch of the selected move and searches for alternative moves of high value and suggests moves to the player for evaluation. This process could be iterated with computer search constrained to ignore moves that have been considered by the player.

1.4 The Conflict between Creativity and Efficiency

"Creativity" is a word often used in a positive sense, but real-life application creativity can also be problematic. As much of this report advocate the implementation of more creative solutions to known problems, this section is added to give a more balanced picture.

There is no doubt that creativity lies behind the amazing development that the human race has brought about the world in the last few thousand years, but perhaps it is also fair to say that the built-in resistance to changes that most of us have, and our tendency of favouring well-tried solutions, have been crucial for our species survival through millions of years. Many decision makers both in civil life and in the military would testify that when there is not enough time to evaluate new innovative solutions, relying on experience is often the best choice. The conclusion we can draw is that the mere fact that an idea is creative does not necessarily mean that it is useful in practice. As a general rule, creative ideas should be properly evaluated before they are implemented. In a way this is contradictive. The obvious risk is that the evaluator is biased and discards an idea for the very reason that it has not previously been tested.

Another contradiction is that when we talk about "creativity" as a desirable trait, there is usually an underlying assumption that the creativity should stay within some bounds. This is especially true when creativity is discussed in relation to a specific application. In order for a creative idea to be possible to incorporate in an existing product or organization, the idea must not be too far out. Also, it must not be creative in the “wrong sense”. A trivial example can be taken from automatic story generation. If an automatically generated story, set in the Wild West during the 19th century, would feature a villain that decided to rob a bank armed with nothing but a rolling pin, it is likely that we would think that the story generation was both creative and successful. On the other hand, if the villain used a .44 Magnum, it is likely that we would not appreciate the creativity of the system placing a 20th century weapon in the Wild West, but deem the story as a failure.

Clearly, there are no easy solutions to the problems discussed above. There will always be a problematic trade-off between discarding new ideas too easily and uncritically adopting untried solutions. If an overall conclusion could be drawn from the above discussion, it would perhaps be that the question of how to support objective evaluation of creative contributions to real-life applications deserves more attention.

1.5 Applications of Computational Creativity

This section provides an overview of current and future applications of computational creativity. Since computational creativity is a young field, most of the applied research is experimental and exploratory with real-life use somewhat in the future. At this point we will not explain how the methods work but rather provide pointers to the literature review in sections 2 and 3.

1.5.1 Decision making

It appears that little computational creativity research has been directly aimed at decision making. This section includes the few examples that we have found in the open literature together with recent research performed at FOI.

The DARPA project Deep Green (Surdu and Kittka, 2008) includes research goals that are very similar to computational creativity but we have not identified anything in the research literature that describes the progress towards these goals. Deep Green aims at enhancing operational military decision making with advanced user interfaces that connects commanders and operational staff to artificial intelligence services supported by highly realistic simulations. The commander sketches operational plans; virtual AI assistants break down the suggested course of actions to a multitude of alternative detailed operational plans that include a wide range of possible enemy actions and submit the panoply of implementation plans to the simulation engines. Simulation results are summarized by the AI and key decision points are submitted for human evaluation. The purpose is to get inside the opponent´s decision loop and increase the pace of operations to a point where the enemy´s command and control capacity collapses. In the computational creativity branch of Deep Green, AI assistants use the vast knowledge base of possible operational outcomes to propose alternative courses of actions for consideration by the commander.

Tan and Kwok (2009) apply computational creativity to finding scenarios for terrorist attacks against the harbour and shipping lanes of Singapore. The scenarios guide development of security measures implemented by the police and the armed forces of Singapore. Examples of scenarios and details on the methods are found in subsection 4.1.2.

FOI has contributed to computational creativity research related to decision making. Jändel (2013a) describes how to extend naturalistic decision-making processes with computational creativity agents. A naturalistic model captures and formalizes how real-life decisions are made. The Recognition-Primed Model (RPD) is a leading naturalistic decision-making model that is guided by studies of how successful leaders in for example business, fire fighting and the armed forces make operational decisions (Klein et al., 1986), (Klein, 2003, 2008). The RPD process recognizes that experienced leaders intuitively evaluate situations and identify what kind of goals that are reasonable to pursue, how the situation can be expected to evolve, what information that is relevant and what courses of actions should be considered. Jändel outlines how computational creativity agents can support the RPD process by providing ideas on

1) novel courses of action,
2) aspects of the situation that should be taken into account when assessing proposed plans,

3) how to repair a promising operational plan that fails in initial mental or computer simulations,
4) situation anomalies that may be unnoticed by decision makers,
5) novel situation assessments that resolves observed anomalies,
6) information that should be considered when assessing the situation.

Examples of computational creativity methods that could be used for achieving these functions are further discussed in (Jändel, 2013a).

Deception and counterdeception are important aspects of human conflicts and notoriously difficult to handle in decision making. Coming up with a new deception strategy is an act of creativity. The intended victim´s uncovering of deception requires even more creativity. Jändel (2013b) argues that advanced information fusion systems are vulnerable to deception and shows how computational creativity methods can be recruited for the purpose of building automatic or semi-automatic counterdeception capacity. It is found that technical methods for fortifying an information fusion system with counterdeception resourcefulness often are analogous to the creativity techniques that are used in human teams.

The research field of automatic story generation is focused on creating new and innovative stories from a *knowledge base* containing story fragments, plot skeletons, context information, etc. One of the main challenges is to construct stories that are logically consistent, so that for instance causal relationships are preserved. In existing literature, most story generating systems are configured so as to produce stories that resemble traditional fairy tales and fables, but the techniques for automatic plot generation could equally well be used to generate more realistic scenarios. The idea we propose is that systems for automatic story generation should be considered for use in decision support. Given appropriate constraints and background knowledge, a story generating system could be used to automatically generate a variety of scenarios that show different possibilities for how a situation could develop. Without the prejudices of a human, an unbiased automatic system may be able to foresee a possible turn of events that is logically feasible, but that a human analyst might overlook because it is not expected. Some story generating systems, like the classical Tale-Spin

(Meehan, 1977) are problem centred, *i.e.*, designed with the objective to create a story in which a given problem is solved. Though the problem solving capacity in existing systems is not impressing, these systems can serve as inspiration for the development of systems that have more advanced problem solving abilities and that could be used as support for planning on strategical or tactical levels.

In spite of the scant and very recent literature on computational creativity for decision support, we think that it should be quite feasible to build truly useful applications without developing any fundamentally new techniques. The main task in computational creativity is to provide novelty combined with relevance for the situation at hand. Methods for creating novelty are provided by the present tool-box of computational creativity and decision making is actually quite unique in that simulation tools for automatically evaluating relevance often is available.

1.5.2 Training

In the last decade, the use of video games and simulators in military training has rapidly increased. Video games provide a comparatively cheap and versatile complement to traditional full scale military exercises. Until now, video games and simulators have primarily been used to train soldiers' skills in combat – for instance, the US Army uses a version of the commercially available video game *Battlefield II*, which is a traditional *first person shooter* game – but there is a large potential in using video games to train other skills as well. The next versions of the US Army training simulator is planned to include functionality that help soldiers train cultural awareness, language, combat lifesaver skills, etc.[1]

Military training simulators of today are generally built on the same technology as commercially available video games for entertainment. This is in one way problematic. The development of video games for entertainment is driven by what is predicted to sell, and unfortunately the economic incentives to create varied and unpredictable gaming experiences for the players are not necessarily very large. Today's

[1] http://www.huffingtonpost.co.uk/2012/08/02/us-army-video-game-training_n_1731816.html (page accessed on 2013-02-21)

video games are to a large extent predictive, or more precisely, only a limited number of things could happen. For a soldier training to meet the complex reality, for instance on a mission in Afghanistan, this is unfortunate. By adding more creative ability to the game generator, the training experience could be significantly improved. In fact, military training is probably one of the applications in which the potential contribution of computational creativity is easiest to envision. In the rest of this section we discuss some concrete examples on how computational creativity could be integrated in video games and simulators for military training.

In a video game or a simulator intended for military training, it is desirable that the trainee should be able to react and act in the virtual world as freely as possible. At the same time, it is desirable that the training command should have some means to target the training by influencing the sort of situations that the trainees encounter. Finding a solution that satisfies both requests at the same time is non-trivial. If the premise is that the trainee's actions should affect the happenings in the virtual world, pre-planning of the plot is not an option. Instead, desired events must be introduced in the plot by an adaptive planning algorithm that can at least make sure that the pre-conditions for an event are satisfied before the event is launched. To make events fit smoothly into the plot, some amount of creativity is also needed. Techniques for how to treat the case that either a human or an external computer program interferes with plot planning can be found in literature related to the research area of *Automatic story generation*, where this sort of problems frequently occurs (Peinado et al.,2005), (Swartjes and Vromen, 2007), (Perez y Perez et al., 2010).

In video games in general, and first person shooter games in particular, characters that are not played by real persons are run by a very simple AI logic. Again, the economic incentives have not been sufficiently strong to promote development of more advanced AI-based characters. In military training, more intelligent and creative behaviour from the enemies would force the soldiers in training to be creative as well. A more interesting and less predictive behaviour for AI based characters in training environments could for instance be achieved using Case Based Problem Solving (Swartjes and Vromen, 2007). It would also be possible to go one step further and let the AI-players, in part, be driven by “emotions” and personal goals (Rank et al., 2012) or personal traits such as education and ideology (Pavón et al., 2006). In a game or training simulator with such complex characters, it would not only be possible to train combat skills, but to some degree also social skills that in real life are important in most missions.

The examples in this section have been focused on training in virtual environments, but the reader should keep in mind that techniques for computer supported adaptive plot generation could just as well be used in more traditional military exercises, such as the VIKING exercise[2].

1.5.3 Product innovation and marketing

Inventing new products and coming up with marketing messages that drive home the commercial success of the products are areas where creativity is a key asset and where idea generation is known to be a time consuming and expensive process. Budding branches of the research in computational creativity are hence aiming for automating creativity in product innovation and marketing.

Li et al. (2012) apply computational creativity to inventing gadgets that fulfil requirements provided by the user. The first application is to generate fictitious devices that are needed in automatically generated stories. The storyline provides the requirements for the gadgets and the task of the algorithm is to invent a device that both fulfils the requirements and is fun and interesting. The story about Harry Potter requires for example a device that allows the orphaned Harry to meet his deceased parents. The Mirror of Erised satisfies this purpose but includes also many other details and functions that are added to make the story more interesting. The methodology of Li et al. is described in subsection 4.1.3 and could easily be modified to support invention of real devices.

McCaffrey and Spector (2011) describe Innovation Assistant software that has been applied for example to finding new methods for road-side bomb detection. Recognizing that people often fail to notice the

[2] http://folkebernadotteacademy.se/sv/Verksamheter/Utbildning-och-samtraning/Expertstod-till-ovningar/VIKING/ (page accessed on 2013-02-21)

features in the situation that are most relevant for finding a solution, their method is designed to automatically provide such crucial insights to human product innovation teams. See subsection 3.3.2. for a full account of the Innovation Assistant.

Strapparava et al. (2007) apply computational creativity to automating creative work in advertising. Subsection 3.3.1 provides further details on their work. The key idea is that the user enters information that defines the context in which an advertising idea should be generated and that the system harvests a repository of background information for ingredients that are moulded into a list of ideas intended for human assessment and elaboration. This research direction could find future applications in psychological operations.

2 The Research Field of Computational Creativity

This is the first chapter of the literature review where we very briefly summarize the foundational discussion about what creativity means and in what sense machines can be creative. The history and present characteristics of computational creativity and its demarcation to neighbouring research fields is furthermore delineated.

2.1 What is Creativity?

Creativity is one of those concepts that are notoriously hard to define. The research community is split between fundamentalists and pragmatists where the former believe that it is impossible to work in a scientific way without a precise and measurable definition of creativity while the latter hold that it is better to successively demonstrate artificial creativity in as many domains of application as possible and use this experience to continuously refine an operational definition of creativity. The rift between fundamentalists and pragmatists is aggravated by the philosophical, religious and metaphysical connotations of creativity. The concepts of creativity, intelligence, morals and emotion are very closely related to what it means to be a sentient being and to have human value. Some philosophers argue that it is impossible for machines to be creative since creativity requires consciousness and only humans can be truly conscious. Others hold forth that human intelligence including creativity, physically is caused by signalling between the nerve cells in the brain and that computers in principle should be able to mimic the signalling patterns of the brain and thus achieve both creativity and any other human mental faculty.

Since there is no generally agreed precise and measurable definition of creativity, we will in the following apply a rather loose operational definition according to:

Creativity is the ability to generate novel and valuable ideas, concepts and artefacts where novelty and value is adjudicated by some receiving human or group of humans.

This definition means that creativity is a subjective property of ideas and that the level of creativity is appraised by some receiver that may be a single user, a small group of users, an organization, a society or humanity at large. A creative idea must be novel and valuable. Novelty is evaluated relative the knowledge of the receivers and value is appraised according to the goals, needs and taste of the receivers. Creativity does hence not imply any metaphysical quality of the originator, total originality of the creative product or unanimous acceptance by all critics.

Boden (1992) defines three main types of creativity:

- Exploratory creativity
- Combinatorial creativity
- Transformational creativity

Boden's classification scheme may not be entirely well-defined from a computer science point of view but it is widely accepted, frequently used in the literature and convenient as a provisional nomenclature.

Exploratory creativity searches for novelty and value within a predefined search space. Using Paris taxis for transporting troops in the first battle of Marne (1914) was the result of creative explorative search in the space of *means of transport*. The search space may in some cases be very large. Much of architecture is for example about exploring the space of arrangements of building elements, which may result in very creative buildings such as the Sydney opera house.

Combinatorial creativity blends familiar ideas, concepts and things into new and valuable combinations. The user interface design of the first generation of personal computers can for example be viewed as a blend between a TV and a typewriter. Combinatorial creativity is common in literature and language for example in the form of analogies and jokes.

Transformational creativity changes the boundary of the search space and what types of things that can be combined. Aircraft design was inspired by birds and can be understood as searching in a design space encompassing, among many other things, the aerodynamic shapes of wings. Transformational creativity was required for inventing the helicopter where lift is not provided by wings.

The differences between Boden’s categories are not always clear-cut from the perspective of computer science where combinatorial creativity could be viewed as exploratory search in the space of combinations and blends. Transformational creativity could furthermore be construed as explorative search in a metaspace of search spaces. In spite of these concerns, we will use Boden´s canonical scheme as a provisional taxonomy of creativity.

2.2 Relations to other fields of research

Computational creativity is an incipient science and there are no general agreement on its precise objectives and boundaries. For this literature study, we will use the following makeshift mission statement:

The objective of computational creativity research is to generate knowledge that enables and explains creative computer programs. This knowledge is used in various application domains and for explaining natural creativity. The methodology of computational creativity is to apply computer science inspired by cognitive science, psychology and neuroscience.

Creativity is an aspect of intelligence and computational creativity is therefore a branch of Artificial Intelligence (AI). There are two main research issues in both of the fields: learning how creative/intelligent machines can be built and understanding how human and animal creativity/intelligence works. Both computational creativity and AI are therefore in the confluence between the larger fields of computer science and cognitive science where the former is the science about computation in a very broad sense and the latter mainly is the science about biological cognition although there are significant overlaps. Computational creativity is hence a very cross-disciplinary field with relations to psychology, philosophy, neuroscience and many other fields.

Idea processors are used in business, engineering and decision making for organizing and archiving products of human creativity (Chen, 1998). User generated ideas are entered via an interface and organized, stored and visualized by the idea processor. In contrast to idea processors, computational creativity is required to autonomously spawn creative ideas rather than just provide a framework for human creativity. We will therefore not further consider idea processors in this report.

Computational creativity investigations can roughly be divided into foundational research and applied research. The former direction focus on how creativity is best understood in terms of computer science and how creativity rigorously can be measured and evaluated. The applied branch focuses on demonstrating creative computer programs in product innovation, marketing, story generation, poetry, music, painting and other visual arts. FOI is presently working towards a wider application of computational creativity in decision support.

The methodology of computational creativity is a reflection of the twin objectives of novelty and value. It is often easy to automatically generate a very large set of novel ideas. The main problem is to filter the torrent of machine generated novelty for ideas that are valuable from the user’s point of view. As a simplistic example consider the problem of generating an original name for a new brand of products. A computer can easily provide long lists of randomly combined letters where nearly all items are completely new but also quite useless for human receivers. Designing artificial intelligence to filter the random rubble for gems of creative value is much harder since the AI would need some understanding of the business context of the company and the preferences of its customers.

Novelty in computational creativity is often achieved by some form of search in representation spaces using well-known computer science techniques including logic, planning, genetic algorithms, genetic programming and multi-agent programming. Chapter 4 in this report describes computational creativity research from the perspective of computer science methods.

Methods for ensuring value are much more application-dependent since the operational paradigm puts the human receiver as the final arbiter. This is particularly problematic in the arts where value is judged relative the often very extensive traditions of a field of art according to the fleeting fads of art critics. The problem of filtering for value is in fact much more benign in decision support where computer simulations of the situation can be applied for automatic evaluation of machine generated suggestions. Chapter 3 in this report views the field from an applied point of view and discusses how value is judged in various domains.

Computational creativity became an organized and recognized branch of artificial intelligence research about ten years ago. A computational creativity workshop was held yearly from 2003 to 2006 hosted by AI conferences. Since 2007, there has been a yearly dedicated event, first under the name of the International Joint Workshop on Computational Creativity and from 2010 as the International Conference of Computational Creativity. Several special scientific journal issues have covered computational creativity but there is no dedicated journal. U.S. and U.K. science funding agencies have earmarked funding for computational creativity and EU organized 2011 a special seminar on computational creativity in preparation for upcoming research funding calls (Creativity and ITC, 2011).

3 Applications

This chapter reviews the computational creativity literature from the perspective of applications with a strong focus on automatic story generation which is one of the most researched applications. We will also explain the methodology that is used in the selected applications.

3.1 Automatic Story Generation

In this section, we will consider a subfield of computational creativity known as *story generation*. Automatic story generation is in a way closely related to generation of other sorts of creative art (McCormack and d'Inverno, 2012), but while the contribution of computational creativity, in for instance, poetry or music composition is evaluated solely on the aesthetic qualities of the output, story generating systems can have practical applications and are thereby subject to other evaluation criteria.

The reason for looking specifically on Story generation in this report is that we see a number of potential applications for this type of computational creativity in both military and non-military decision support and training. We will soon return to the subject of applications, but first we will discuss the concept of story generation and review some of the existing literature on the subject. As the research area of story generation is vaguely defined, we have concentrated the work in this section on reviewing the research presented from 2007 and forward, in the *International Conference on Computational Creativity* (ICCC) and its predecessor the *International Joint Workshop on Computational Creativity*.

The term *story generation* is generally used to denote the area of work dedicated to the automatic (or semi-automatic) generation of “stories”. The term is unfortunately confusing and has been associated with generation of all sorts of narrative components, from characters and settings to plots, and – last but not least – the actual text or speech used to tell the story. Chatman (1978) proposes a taxonomy that defines the relationships between different narrative components (see Figure 3). Chatman uses the term *Narrative* rather than *Story* to denote the root element of the taxonomy. On the second level of the taxonomy, Chatman distinguishes between *Story* and *Discourse* and lets *Discourse* cover everything related to text and speech generation. From a technical point of view, automatic generation of text or speech differs significantly from generation of other narrative components. It therefore seems appropriate for our purposes to make a clear distinction between *story* (including characters, settings and events) and language synthesis. Given state-of-the-art, our belief is that techniques for generation of *stories* are more readily applicable in military systems than are techniques for text generation/discourse. In the rest of this section we will focus on the generation of stories and story fragments, in accordance with Chatman's taxonomy, and disregard the discourse.

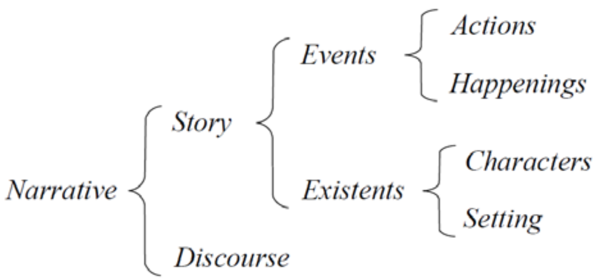

Figure 3. Chatman's taxonomy of narrative components (Chatman, 1978).

3.1.1 Approaches to story generation

All story generating systems are depending on some sort of background knowledge about the nature of the world. The knowledge provided as input to the system is sometimes referred to as the system's *knowledge base.* A new story or story fragment is typically created from transformation and/or mixing of story fragments that are found in documentation available to the system through the knowledge base. In theory, the *knowledge base* that is used by a story generating system could consist of regular text documents, images and sound recordings, etc. In practice however, input to the system is practically always manually pre-processed and transferred to a data format and a representation that can be directly used by the system. An example of a representation form is *logic predicates*, used for instance in (León and Gervás, 2008).

One of the great challenges for automatic story generating systems is to make sure that the stories produced “make sense”, *i.e.*, they are coherent and complying with the “rules of the world”. For example, it would not in general be acceptable for a character to die in one scene and reappear in the next scene eating an apple. Rules that we normally expect to apply include the laws of physics and a variety of causal relationships. If the rules are not stated explicitly in the knowledge base, they must at least be possible to infer from the available information. One way of making sure that basic rules are not violated in story generation is to apply planning techniques. Another approach is to generate stories through simulation in such a way that events can only be triggered if all associated pre-conditions are satisfied. *Planning-based story generation* and *simulation-based story generation* can be seen as two main directions in the literature on story generation. In this subsection, we describe the different approaches and discuss their most distinctive properties. Of course, many existing systems use a combination of planning and simulation techniques in order to improve performance.

Planning-based story generation

A common approach to story generation is to define a fixed set of characters and events and then combine these characters and events to make up a plot. Clearly, the events cannot be randomly distributed if we want the story to make sense. A fundamental constraint is that causal relationships are satisfied, *i.e.*, the events must be arranged in such an order that implicitly or explicitly given pre- and postconditions for each event match the rest of the story. This can be seen as *hard constraints*, as they are logically motivated. We illustrate with an example from (Gervás et al., 2007).

Example: *Gervás et al. (2007) describe MEXICA – a system for automatic story generation. In MEXICA, a set of actions is pre-defined and the pre- and postconditions for each action are explicitly given. An example, in which characters have been inserted, is shown in* Table 1.

MEXICA uses sample stories from the knowledge base to find out in which order actions normally follow each other. The story is partly driven by emotional links *between the characters.*

Precondition	Action	Postcondition
The hunter hates the knight	Hunter killed Jaguar Knight	
	Princess decorated Eagle Knight	The knight is very grateful towards the princess
The knight must be injured or killed	Princess healed Jaguar Knight	The knight has been cured The knight is very grateful towards the princess

Table 1. Example of pre- and postconditions in MEXICA (Gervás et al., 2007).

If pre- and postconditions (or a corresponding set of "rules") are explicitly defined, the story generating system can be made to automatically check compliance when new events are added to a story. In MEXICA (Gervás et al., 2007), compliance is regularly checked during the so called *reflection phase* of the story generating process and if necessary, additional actions are added to the story so as to satisfy remaining constraints. Riedl (2008) presents a similar but more general approach, where a type of agents called *critics* check the story for flaws. Each critic is specialized to detect and repair a certain flaw, where a flaw in this case could be any violation of given rules.

The planning approach makes it possible to verify that a generated story is consistent, but a consistent story is often not enough to satisfy the audience. Depending on the context there may be additional requests, such as that the story should have a happy ending or pass through some given intermediate states. Such constraints could in some cases be seen as hard constraints, while in other cases they should rather be considered as *soft constraints*, or *preferences*. Adding some constraints that make "exciting" events happen and force the story to reach some sort of conclusion tend to improve the end result from an aesthetical point of view. In the military applications where we see that story generation could be used (see Section 1.5) it is also apparent that some means of controlling the story are wanted; when storylines are generated for use in training one would like to be able to influence what events should be included, and in decision support the possibility of specifying both start and end state in series of consecutive events would for instance make it possible to find alternative ways of action that could lead to the same goal.

The task of the planning algorithm is to arrange events in an order where as many preferences as possible are satisfied at the same time as no "hard" constraints are violated. Thus, the planning problem can be formulated as an optimization problem,

Min/max *# preferences satisfied*
while *all hard constraints satisfied.*

Depending on the problem, it may be possible to find a solution by applying standard optimization methods (Griva et al., 2008). However, even if the problem should be theoretically solvable, optimization algorithms are often time consuming and computationally heavy. If optimality is not critical for the intended application, it may be better to spend resources on finding a "sufficiently good" feasible solution. One example of this is the story generating system Talespin (Meehan, 1977). The knowledge base of Talespin included a set of pre-defined characters, a set of pre-defined "problems" and a set of pre-defined events with associated pre- and postconditions. When the system was given a character and a problem for the character to solve, it was designed to search for a feasible solution to the given problem. With this approach, Talespin was able to generate some quite appealing short stories, although the success to a large extent was due to the fact that the search space was small and controlled. An example from (Meehan, 1977) is given below.

Example:

Characters: *George Ant and Wilma Bird*

Problem: *George is thirsty*

Excerpt from story: *George was very thirsty. George wanted to get near some water. George walked from his patch of ground across the meadow through the valley to a river bank. George fell into the water.*

George wanted to get near the valley. George couldn't get near the valley. George wanted to get near the meadow. George couldn't get near the meadow. Wilma wanted George to get near the meadow. Wilma wanted to get near George. Wilma grabbed George with her claw. Wilma took George from the river through the valley to the meadow. George was devoted to Wilma. George owed everything to Wilma. Wilma let go of George. George fell to the meadow. The end.

It should be pointed out that planning-based systems like Talespin are normally not designed to find *all* possible solutions to a problem but to find *one* feasible solution. If the story world (the *universe* and the number of *constraints* defining the *conceptual space*) is small, it may be possible if required to find all solutions using extensive search algorithms. However, if the world is more complex, all solutions may not be possible to find in limited time. In case the planning process is non-deterministic, *i.e.*, it contains some random elements that prevent the system from always generating exactly the same solution for a given setup, then at least some alternative solutions to a problem can be found by running the planning process several times.

A fundamental problem with planning-based story generation is that it is not possible, for an arbitrary problem, to guarantee the existence of any feasible solution at all. This problem remains unsolved.

Examples of planning-based systems are found in (Gervás et al., 2007) and (Riedl and Young, 2005).

Simulation-based story generation

An alternative approach to story generation is to set up a world peopled with individuals, define all the rules that apply in that world, for instance rules that decide how characters in the world interact, and then run the story generative process as a simulation from some chosen initial configuration. The main difference from the planning strategy that was described above is that, in this case, no attempts are made to control the content of the story once the general rules that define the world are set. As a consequence, the generated story will automatically comply with the given rules and no additional planning steps are needed. On the other hand, one may have to run a simulation several times to get some interesting output, and in general the storyline will not have a natural ending that concludes the story. Still, if the output is intended for use in decision support or training, a "loose end" may not be a problem. Simulation-based systems are described in (Theune et al., 2004), (Pavón et al., 2006), (Hassan et al., 2007) and (Swartjes and Vromen, 2007).

As events and actions in simulation-based systems are primarily induced by the individuals populating the story world, simulation-based systems tend to have a focus on *characters* and their driving forces, while planning systems on the other hand tend to focus more on *events*. This means that in simulation-based systems, more effort is generally put into creating believable characters. In addition to other properties and capabilities, characters can be equipped with "emotions", "affections" and personal goals that influence their actions. Possible configuration choices for such characters are discussed in (Rank et al., 2012). A common approach is to let the characters be driven by a combination of *belief* (what the character believes about the world), *desire* and *intention*. It is also possible to let characters be affected by *emotions*. While beliefs, desires and intentions can be modelled as more or less static, *emotions* are constantly changing as a result of the character's interaction with the story world. Emotions are therefore more difficult to model and one may have to consider such things as *creation thresholds* and *decay rates*.

As far as possible applications are considered, a strong advantage of simulation-based systems, compared to planning-based systems, is that real-time interaction of an external player or operator is much easier to realize. Characters that are played by real people can be treated in analogy with characters that are run by AI-algorithms and no re-planning is needed. From research on interactive simulation-based story generating systems, the step to game design and virtual training systems is small. Creativity in game design is discussed further in (Smith and Mateas, 2011), and design of training systems as a possible application of story generation techniques was discussed in Subsection 1.5.2.

3.1.2 Creativity in story generating systems

The two types of creativity (see Section 2.1) that are regarded as most relevant for the *conceptual space*[3] are *Exploratory* and *Transformational* creativity. A typical example of exploratory creativity applied to story generation is the creation of new stories by mixing and re-arranging elements – such as characters, motives and events – from existing stories. Several examples of systems that primarily use exploratory techniques in the generative process were presented in Subsection 3.1.1. From a technical point of view, exploratory creativity is easier to realize than transformational creativity. In return, the output tends to be more predictive and less original.

Transformational creativity, by definition, requires that the conceptual space is changed or transformed by the addition of new story elements or new combinatorial possibilities. This requires more refined methods for data processing than was needed in the case of exploratory creativity. In (León and Gervás, 2008), transformation of the conceptual space is achieved by occasionally allowing some of the rules that decide how the world works to be violated. In other words, some *meta-rules* are used to modify the existing rules. Standard methods for transforming the conceptual space include the use of *Case-based reasoning* and *Ontologies*, which are described here. See also (Zhu and Ontañon, 2010) for more related reading.

Case-based reasoning (CBR)

Case-based reasoning should be seen as a framework for problem solving. The idea is to apply experience, gathered in a knowledge base, to new problems. In the case of story generation, the knowledge base would ideally consist of a set of already existing stories or story fragments. The basic procedure for Case-based reasoning is summarized by the following steps (Aamodt and Plaza, 1994):

1. **Retrieve:** Given a target problem, try to find a similar problem with an associated solution in the knowledge base.

[3] The abstract location of the entities produced by creative processes (León and Gervás, 2008).

2. **Reuse:** Map the solution from the similar problem to the target problem. This may involve adapting the solution so as to fit the target situation.
3. **Revise:** Test the new solution and, if necessary, revise.
4. **Retain:** When the solution has been successfully adapted to the target problem, store the resulting experience as a new case in memory.

Swartjes and Vromen (2007) give an example of what the process might look like when applied in the area of story generation.

Example: *A system is given the task of generating a story in which a knight commits suicide. The problem description given to the system as input is: "A knight does something that results in the knight's death". The problem is solved as follows.*

1. ***Retrieve:*** *The system searches for a similar problem in which the knight does something that results in somebody's death. The following solution is retrieved:*
   "A knight fights a troll with his sword, killing the troll and being injured in the process"
2. ***Reuse:*** *After adaptation the system produces a new solution:*
   "A knight fights and kills himself with his sword"

To verify that none of the rules that define the story world are violated, these two steps should be combined with an evaluation procedure, such as the one described in (Riedl, 2008).

Ontologies

An *ontology* is a model that formally and explicitly defines how different concepts within some domain are related to each other. Typically, an ontology consists of a collection of relations and logical rules. An example of a logical rule is the following statement (Franke et al., 2012):

```
(x of_type Surface_vessel) AND (x transports Persons)
→ (x of_type Ferry)
```

Ontologies can be used in story generation to transform existing story fragments by replacing one concept or attribute with another one that is sufficiently similar. To illustrate this, we use an example from (Swartjes and Vromen, 2007).

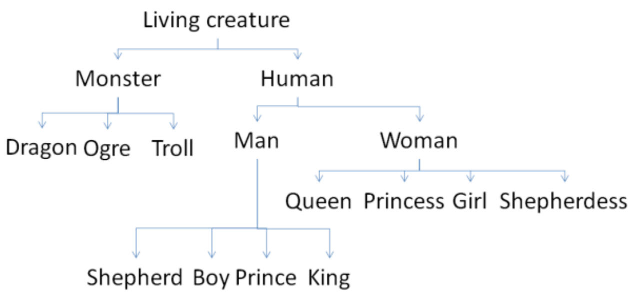

Figure 4. Taxonomy.

Example: *Assume that an ontology consists of the above taxonomy that describes relationships between some concepts. Consider the story fragment "A princess runs away from a dragon". Generalization of concepts using the given ontology could lead to the following transformations of the original story fragment:*

- *"A princess runs away from a monster", which in turn could lead to retrieval of cases about princesses running away from ogres and trolls.*
- *"A woman runs away from a dragon", which could lead to retrieval of cases about queens, shepherdesses or little girls running away from dragons.*

If the ontology had also stated that run *was a special case of the more general term* move *then the output might have become even more interesting:*

- *"A princess moves away from a dragon", which could lead to retrieval of cases about princesses walking, sailing or swimming away from dragons.*

For more reading on ontologies, the reader is referred to (Franke et al., 2012).

Evaluation of creativity

As discussed above, transformational creativity is generally considered more "truly" creative than exploratory creativity. However, there is no consensus about how to objectively evaluate the "creativity" of a story. For example, Gervás and León (2010) suggest that the result of exploratory processes can compare well with results from transformational processes in experienced creativity. A contributing reason is that the amount of "creativity" wanted in a story is closely linked to the intended application of the story. It is in part because of this that the overall quality of a system's output, including the creative contribution, is traditionally judged manually based on a selection of the output (Zhu, 2012).

3.1.3 Applications of automatic story generation

Most of the obvious applications for automatic story generation are within entertainment. The computer game industry, for instance, could use many of the methods developed in computational creativity, perhaps in particular methods originating from the field of automatic story generation (Smith and Mateas, 2011). When considering potential application areas that are of interest for the Swedish Armed Forces, we particularly want to point out *decision support* and *training* as two areas that could benefit from integration of methods from story generation. These applications were described in more detail in Section 1.5.

3.1.4 Interesting research directions

On long term, the aim of story generating research is to build systems that can automatically produce a complete *Narrative* (Figure 3), including characters, a world in which the story is set, and a coherent storyline that contains some twists or an interesting chain of events. Last but not least, the story should be perceived as novel by the reader/listener. State-of-the-art today is far from that goal, but some areas of the research field have come further than others. The areas that we see as particularly interesting for the Swedish Defence Materiel

Administration (FMV) are those that have both of the following characteristics:

a) They have interesting applications for the Swedish Armed Forces AND
b) They are relatively mature and are not very far from being possible to integrate in applications, although much work may still be required before the full potential is reached.

The development of algorithms that make AI-agents in training systems behave more realistically is an example of an area that we identify as interesting. The motivation is that we believe that a relatively small effort in this area would significantly improve the performance of today's systems. Another interesting area is human-system interaction, in particular the problem of automatic on-line re-planning of a training scenario as a response to the actions of a human player (Peinado et al., 2005).

The planning-based story generating systems that have been reviewed in this study seem to focus on *solution principles* rather than on *solution efficiency*. The computational complexity and scalability of search and optimization algorithms has not been crucial since the solution spaces for the problems have been comparatively small. Most likely, the methods need to be improved in terms of efficiency before they can be used for more extensive real world scenarios, for instance in decision support.

3.2 Other applications

3.2.1 Advertising

The default reference for this subsection is (Strapparava et al., 2007).

This work aims at automating the work of copy writers and art directors in advertising by application of natural language processing and text animation techniques. The output of the system is a list of advertising messages intended for further selective processing by advertising professionals and eventually presentation of a subset of the list to the customers of the advertising agency. The main input to the system is a list of familiar expressions, which in the paper is exemplified by a list of well-known movie titles that are harvested from the Internet. The system searches for variations of the familiar expressions in which novelty is created by replacing one of the words in the expression. This is an application of the *optimal innovation hypothesis* (Giora, 2003), which states that pleasant surprises are generated by combining familiarity with novelty in the communicative message. This principle illustrates that maximal creativity in many applications does not imply maximal novelty but rather the right measure of novelty in an otherwise familiar context.

The system applies several clever mechanisms to restrict the search to potentially valuable contributions. An agreeable phonetic impression is created by restricting the new word to assonances of the replaced word using a machine-readable pronunciation dictionary. The movie title *Saturday Night Fever* might e.g. be modified to *Saturday Fright Fever* according to the assonance principle. Aiming for messages that are emotionally arousing, the search will also give priority to replacements with high emotional valence. This is achieved by using and extending the WordNet-Affect database where words are annotated with emotional labels. Words that lack an intrinsic emotional charge inherit the valence of emotional terms that are found to be statistically related to the target word in a large corpus of documents. The intrinsically non-emotional word “university” is for example, by the application of statistical natural language processing, found to be related to the emotions of *Enthusiasm*, *Sympathy* and *Devotion*. Finally the system will select fonts and modes of animation that are consistent with the emotional charge of the message

As an example of system output, an advertiser could promote shopping in the London Notting Hill district using mash-ups of the *Notting Hill* movie title. Selecting for the emotions of Exhilaration and Gladness the system would deliver *Notting Thrill* and *Notting Chill* respectively. This work illustrates that a computational creativity system can use knowledge bases to automatically filter the results of random search so that the final output is found to be both creative and useful. Creative message generation is, in particular, facilitated by extensive open-source databases for natural language processing.

3.2.2 The Obscure Features Method and Bomb Detection

The default reference for this subsection is (McCaffrey and Spector, 2011).

The Innovation Assistant software of McCaffrey and Spector (2011) helps humans to solve insight problems using the *obscure features method*. Insight problems are the kind of problems that seem to require human creative intuition. Psychological research indicates that the crucial insight often is related to noticing a feature of the problem that people normally do not pay attention to. Consider for example, the *Two rings problem* where the task is to physically connect two steel rings using only a candle, a match and a small steel cube. Candles consist of wax and a wick. A wick is a string. Once we have noted that the problem inventory includes a string, it is easy to solve the problem by using the steel cube to remove the wax and tying the rings with the wick.

The components of the problem and their attributes can be described by semantic networks which are computer readable descriptions of the concepts, features and relations in the problem domain (see **Error! Reference source not found.** for an example). When humans are asked to draw semantic networks they will typically only find a limited set of commonly noticed features. Most people will for example note that a candle consists of wax and wick resulting in a semantic network where the concept *candle* is connected by the relation *consists of* to the features *wax* and *wick*. Using computerized methods it is, however, possible to generate much larger semantic networks including networks where the feature *wick* is connected by the relation *is a* to the feature *string*. Further psychological experiments show that the features that inspire creative problem-solving insights typically are found just one or two steps beyond the box of commonly noticed concepts.

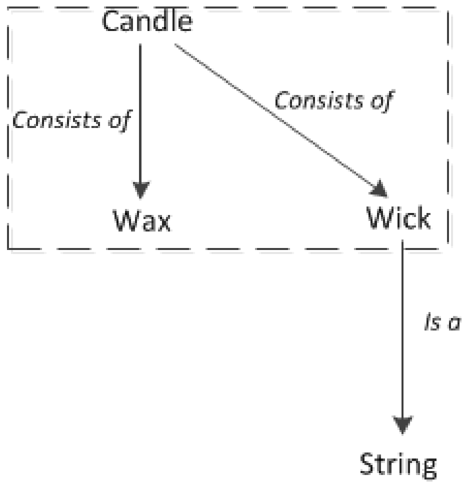

Figure 5. Example of semantic network with dashed box of commonly noticed concepts.

The Innovation Assistant suggests problem space features that humans should pay attention to. Browsing the list of suggested features helps problem-solvers to get creative insights. Building semantic networks using generic ontologies and encyclopaedias is quite feasible but the resulting networks are too big and amorphous. Most of the concepts in the motley mix are unrelated to the problem context. A generic semantic map related to candles will for example include the concepts of *romantic dinners* and the song entitled *Candle in the wind* both of which are patently useless for solving the *Two rings problem*. McCaffery and Spector avoid such bloated semantic networks by fetching the mapped concepts from special taxonomies that are related to the problem domain. For engineering-type problems the semantic networks are restricted to material properties and other physical attributes of the relevant objects. The Innovation Assistant achieves hence novelty by searching a taxonomy for features that humans normally fail to notice and attains relevance by restricting the search to concepts that are relevant for the problem domain and fall at most one or two levels beyond the box of commonly noticed features.

The obscure features method has been applied to road-side bomb detection, industrial product innovation and experimentally to logistics problems. The software is not available yet other as a part of a consultancy package.

4 Methodologies

This chapter reviews the computational creativity literature from the perspective of methodology but exemplifies the methods with relevant applications.

4.1 Research Based on Conceptual Blending Theory

4.1.1 Conceptual Blending Theory

Conceptual Blending Theory (CBT) is originally an explanatory model for how humans think, which recently also has inspired generative algorithms in artificial intelligence research. The psychological theory of conceptual blending is comprehensively described by Fauconnier and Turner (2002) in which further references to the extensive literature on the subject can be found. In this subsection we will briefly review CBT focusing on the relation to computer science.

CBT describes blending operations on *mental spaces* where a mental space is a system of concepts and relations related to some topic. **Travelling** is an example of a mental space that includes the concepts {*Origin location, Goal location, Vehicle, Hardships, Delays, Dangers, People left behind, People encountered on the way*, …} and the relations {*Saying goodbye to, Greeting, Meeting en route*, …}. Humans have a rich set of associations related to many mental spaces including the **Travelling** space.

We use mental spaces literally but also in metaphors where for example the human life is compared with elements from the **Travelling** mental space. Such metaphors are examples of blending operations in which selected elements from different mental spaces are matched and thus can borrow aspects of the rich undergrowth of associations from each other. In blending the **Human Life** space with the **Travelling** space we metaphorically match the beginning of a journey to birth, the edifying hardships of travelling to the privations of human life and the happy arrival to death. This particular blend might be designed to provide solace in face of the harsh realities of life. A large part of the literature on CBT, not discussed here, focuses on explaining phenomena in psychology, linguistics, social science and artistic expression as the result of the blending of mental spaces.

The concept of mental space can be mapped to similar but more rigorously defined concepts in computer science such as *taxonomy* and *ontology*. A taxonomy organizes the concepts of a domain in categories and subcategories. Living organisms are for example organized in a biological taxonomy including hierarchical levels such as Family, Genus and Species. Ontologies extend taxonomies with information about attributes, relations and rules.

Some of the rather loosely defined blending operations that have been suggested in the psychological CBT literature can be expressed as formal merging operations on computer readable taxonomies and ontologies. The *double-scope blending* operation shown in Figure 6 has been found to be particularly useful.

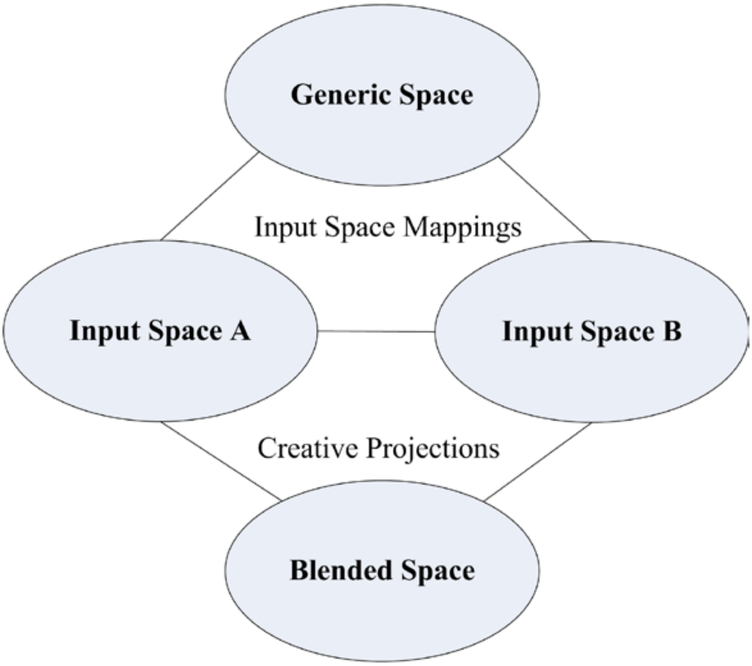

Figure 6. Double-scope blending. Concepts in input spaces A and B are connected by reference to the generic space. Different blended spaces can be constructed by creatively projecting and relating elements from both input spaces. The blended space may be elaborated by pattern completion, associations and specializations.

A simple example of double-scope blending is the analogy between war and surgery. The generic mental space is **Cleansing** in which we find the concepts of an agent performing the cleansing, an instrument of cleansing, something that is removed by cleansing and something that is purified by cleansing. From the **War** and **Surgery** mental spaces we map the Commander in chief and the Surgeon respectively to the cleansing agent; the enemy and harmful flesh respectively to substance that is to be removed by cleansing etc. In the blended space, we can now construe the commander as a surgeon wielding his army as a scalpel to save the invaded country from the cancer of heretical beliefs. Note that different blends can produce radically different results such as for example portraying an incompetent surgeon as a brutal general violating the pristine country of the body with senseless destruction. The act of creation in double-scope blending consists of selecting input spaces and a generic space, using the generic space for matching some of the concepts in the input spaces, project selected elements from each input space into the blended space and finally defining new relations between the concepts of the blended space.

It should be noted that Fauconnier and Turner´s theory originally was intended for explaining observed creative behaviour and is not a generative computational model. Neither how input spaces are selected nor how mappings are done are specified in CBT. Furthermore, there are no defined procedures for how to elaborate the blended space.

4.1.2 Maritime Scenario Planning with CBT

The default reference for this subsection is Tan and Kwok (2009).

Tan and Kwok (2009) describe how double-scope blending is applied to scenario generation for the purpose of improving the security in the Singapore harbour area. The Singapore port authorities handle more than 60 000 ships yearly and the immediate surroundings include a very important oil refinery and airports. Automated decision support systems are developed for assisting the patrol craft and navy vessels that protect the area. Scenario planning is used for identifying threats as a basis for designing strategies e.g., for selecting ships for inspection.

Tan and Kwok (2009) use implicitly the generic mental space of **Shipping** to connect the input spaces of **Maritime Terrorism** and **Peaceful Shipping**. It is assumed that the agent of maritime terrorism is a single ship or small boat that could be on a suicide mission but could also intend to survive the mission. They demonstrate that it is possible to construct five blended spaces that in previous scenario studies have been found to be relevant by humans. Examples of such blended spaces are a container ship attacking using kinetic energy and a tanker ship with on-board explosives. Figure 7 shows another example where the blended space indicates a fishing trawler attacking an airliner with a missile with the intention of creating massive damage in a nearby oil refinery.

The ability to create such detailed scenarios is based on that relevant knowledge bases enrich the representation by connecting to some of the top-level concepts in the input spaces. A knowledge base relating to weapons and explosives is for example connected to the Harmful Agent concept in the **Maritime Terrorism** mental space. In addition, there is a knowledge base for target types connected to the three different abstract target categories. Detailed scenarios are generated by first creating a high-level blended space using the abstract components of the two input spaces and then selecting specializations of some of the abstract components from the associated knowledge bases. The latter process maps to the blending space elaboration phase in Fauconnier and Turner´s CBT.

The expressiveness of the conceptual blending approach causes a combinatorial explosion of possible scenarios. Tan and Kwok (2009) discuss this problem but offer no solutions other than using human analysts for evaluating automatically created scenarios. They mention the possibility of automatic clustering of blends into aggregated scenario constellations. To do such clustering in a meaningful way would, however, require some level of autonomous capacity for intelligently deciding what kind of clusters that would be helpful in the analytical context. It appears, however, that the automatic blending functions described by Tan and Kwok (2009) have been found to be useful for suggesting interesting scenarios for further evaluation and refinement by staff analysts.

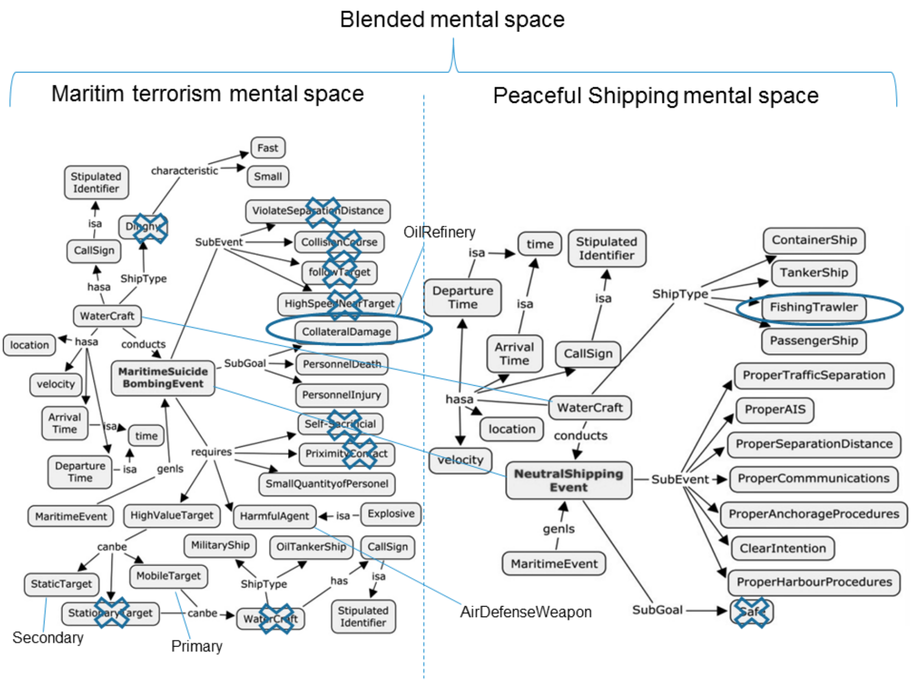

Figure 7. Maritime scenario created by double-scope blending. The dashed line separates the Maritime Terrorism mental space from the Peaceful Shipping mental space. Using the generic mental space of Shipping we map a peaceful shipping event to a maritime terrorism event. The watercraft used in each event is also mapped. A blended mental space is created by exporting all the features of the Peaceful Shipping mental space except for the goal of safe behaviour and the specialization to the ship type fishing trawler. From the Maritime Terrorism mental space, we map everything to the blended space except for the crossed-out elements including for example the sub-goal of self-sacrifice and the behaviours of violating proper separation distances and following a target. We specialize the primary target to be a mobile target of type commercial airplane and the secondary target to be a static target of type oil refinery. The harmful agent is selected to be an air defence weapon. The resulting scenario can be interpreted as a fishing trawler, first behaving normally and then attacking an airplane for the purpose of making it crash into the oil refinery. The scenario is developed by FOI using input spaces from Tan and Kwok (Tan and Kwok, 2009). The figure is also a modified version of figures in (Tan and Kwok, 2009).

4.1.3 Context and goal-driven CBT

The default reference for this subsection is (Li et al., 2012).

Li et al. (2012) use an extension of Fauconnier and Turner´s CBT suggested by Brandt and Brandt (2002) to incorporate context and goals for the purpose of particularizing poorly defined aspects of the original theory. Context and goals drives the selection of input spaces, the mapping between input spaces and the elaboration of the blended space.

Context and goal-driven CBT is applied to inventing hypothetical gadgets for the purpose of enriching automatically generated stories. Light sabres of Star Wars fame can for example be viewed as a blend of a laser and a sword. Humans must supply current computational creativity story generation systems with concepts such as light sabres before they can be used in automatically generated storylines. Li et al. (2012) describe a computational creativity subsystem that autonomously creates novel gadgets, which serve some purpose, defined by the story context.

As an example, consider a Manga story where the goal for a section of the tale is that a character Bob shall be infected by a flu virus carried by another character Alice. Possible solutions are constrained by the fact that Alice is far away from Bob with no means to travel. The context of the Manga genre allows counter-factual and physically impossible gadgets.

Firstly, the algorithm searches the story generation knowledge base and finds a standard story element that normally would have been used to reach the goal (see Section 3.1 for a review of story generation methods). This story element will be the first input space. In this case, it consists of a story snippet where Alice infects Bob by coughing. This particular event is, however, impossible because of the constraint of physical distance. Secondly, the algorithm searches a knowledge base of prototype gadgets looking for a device that can achieve the goal of transmitting something over space. It comes up with the concept of a telephone that will be the second input space. The blended space is a flu-transmitting telephone that is permissible in the context of Manga and achieves the goal of the story fragment.

Details of the gadget-generation algorithm are found in (Li and Riedl, 2011a, 2011b). Goals and constraints are described in first-order logic. Knowledge bases are also expressed in first-order logic and reasoning is performed by a modified planning algorithm. Produced gadgets are tested by the story-generation tool. Gadgets that fail to satisfy the goals and constraints of the story are used as input space to be blended with another prototype from the gadget knowledge base. Several iterations may be required before a successful solution is found.

Li et al. (2012) note that random blends of story elements and gadget prototypes would result in a combinatorial explosion of hypothetical gadgets from which it would be difficult to extract devices that make sense from the point of view of the unfolding story. Guiding the creative process with context, goals and constraints means that a much smaller set of candidate gadgets has to be tested by trial runs of story generation.

The gadget-innovation methodology of (Li et al., 2012) could be interesting in the context of product development where it could provide creative input at various levels of the development process. The first input space could be a set of requirements for a particular system. The second input space would be selected from a knowledge base of relevant prototype systems and the blended space would be a mapping of the prototype to a subset of the requirements enriched with associations and specializations guided by the requirements. The resulting system is tested in simulations and possibly further enhanced in a series of creative iterations.

A set of requirements could for example specify that a small optical sensor on a military vehicle shall be difficult to detect. An important constraint is that the coating of the lens strongly reflects light of visible wavelengths. The computational creativity system finds the concept of fireflies in the camouflage concepts knowledgebase. The blending operation matches the fact that light from fireflies lack infrared components and that the size of the lens is similar to that of a firefly. The resulting innovation is to sprinkle the vehicle with light emitters that are controlled by software to emulate fireflies.

4.2 Information-theoretic approach to computational creativity

Jürgen Schmidhuber directs a world-leading AI group and he also champions a theory that defines creativity in terms of changes in subjective compressibility of an agent´s experiences (Schmidhuber, 2006). To explain this, we shall consider a robot that stores all data that it receives from its sensors and continuously tries to compress the stored data as much as possible. Compression means that an algorithm uses regularities in the raw data to produce a coded description of the data that uses fewer bits than the raw data. If the compression is lossless it is possible to decode the compressed data and recover the original data. Lossy compression algorithms discard information based on some heuristic assumption on what aspects of data that is important. Since Schmidhuber aims for a generic theory, we must assume that the robot works with lossless compression.

Schmidhuber's robot works incessantly on minimizing the compressed code $C(t)$ that describes its stored raw life history $H(t)$ at time t. Suppose that the robot encounters a new object that is represented by a piece of raw sensory data D. The object might be an idea, a piece of art, a poem, a work of science or anything else. Schmidhuber defines the subjective *interestingness* of the new object as the present rate of decrease in the size of the robot´s present internal compressed description of the object. The interestingness of the object at time t is hence,

$$I(D \mid t) = B(D \mid t\text{-}1) - B(D \mid t),$$

where $B(D \mid t)$ is the number of bits in the compressed version of D at time t. Note that the compression algorithm is based not only on data about the new object but also on the robot's entire life history $H(t)$.The object is very interesting if the compression algorithm just have learnt how to compress it much better so that $B(D \mid t)$ is much smaller than $B(D \mid t\text{-}1)$.

Creativity is strongly related to interestingness. Suppose that the robot views an object that is produced by another agent such as a human. Schmidhuber's robot deems the object to be a product of creativity if it has a high interestingness value. A creative picture appears first as being very complex thus requiring a large number of bits to code after compression. The robot discovers, however, new types of regularities in the picture, which makes it possible to improve the robot´s compression algorithm resulting in large reduction in the size of the compressed coding of the picture. The interestingness of the picture is at this moment large and the robot may utter that the picture is very creative. At some later point in time, the robot may be unable to discover any further novel regularities, which make it impossible to further improve the compression algorithm. At that moment, the robot will find the picture boring and hence not creative. Objects of enduring creative value allow the viewer to continuously discover new layers of interesting regularities.

Schmidhuber claims that this is a generic mechanism for subjective interest, curiosity and creativity that applies to humans as well as robots although the human compression mechanism is hidden and largely subconscious. He substantiates his theory by creating interesting pieces of art demonstrating the confluence of esthetical value and very low algorithmic information content[4].

One might argue that objects that enable the robot to find new patterns in its life history and hence increase the compression of the life history also should have a high interestingness. A work of art that gives people a new perspective on their life should surely be considered to be very creative. The life-changing artefact could be a very short poem that in itself has very little potential for further compression.

We have not found any applications of Scmidhuber's theory in decision support but we suggest the following preliminary extension to decision-making. In decision support, the interesting object to compress would be *S*, the total amount of raw data about the situation at hand. In general the raw data is distributed and found in the brains of a team of decision makers and in various digital repositories. For simplicity, we shall in the following assume that all of *S* is in the head of one single decision maker. A decision maker that understands the essence of the situation will according to the model be able to compress the raw data about the situation efficiently while lack of situation awareness implies poor compression performance. Good creative input from a computational creativity assistant would hence be

[4] http://www.idsia.ch/~juergen/femmefractale.html

an idea that helps the decision maker to compress the situation more efficiently. The take-away of Schmidhuber's theory is therefore that a computational creativity agent should evaluate ideas based on an estimate of how much data compression of *S* can be improved based on the idea. Ideally, the computational creativity agent should use the same compression algorithm as the one in the head of the decision maker. Since this typically is impossible it is an interesting research question to consider how a useful approximation of the compression algorithm can be constructed. What are the key properties of the compression algorithm that must be preserved?

We note the similarity between Schmidhubers theory and information-theoretic anomaly detection (Lee and Xiang, 2001) where it is assumed that situations with anomalies are less compressible than anomaly-free situations. Finding a compressed description of the situation with fewer anomalies and hence a shorter description length can be understood as an act of creativity. McGregor (2007) discusses also the relation between novelty in the context of creativity and generic algorithmic compression concluding that it appears to be impossible to define an observer-independent formal novelty measure that is relevant for practical applications of computational creativity. This corroborates Schmidhuber's observer-dependent approach to defining creativity.

4.3 Genetic programming

All problem solving can in computer science be construed as searching for the solution in some space of solutions. Genetic programming searches the space of computer programs using biologically inspired methods. In this section, we first explain genetic search methods followed by a specialization to search in the space of computer programs. After this introduction to the methodology, we explore how genetic programming is used in computational creativity. Since the seminal work of Karl Sims (Sims, 1991) has been reviewed elsewhere (Jändel, 2012), we focus on two more recent applications. We note that computational creativity by genetic programming is the methodology that comes closest to achieving Boden´s transformational creativity.

4.3.1 Genetic search

Genetic and evolutionary methods are search methods that are inspired by how natural evolution works. The solution is encoded in a data structure that corresponds to the *genetic code* of an organism. The search starts by randomly generating a population of genetic codes. The solutions that are represented by the codes are evaluated against some optimization criterion. Poorly performing genetic codes are discarded. The best genetic codes are allowed to reproduce and generate the next generations. This process is repeated for many generations until a good solution has been found. The reproduction process ensures that the genomes of the new generation are based on the most successful members of the old generation but with deliberately introduced random variations. Reproduction algorithms usually include at least two processes corresponding to sexual reproduction and to biological mutations, respectively.

As a simple example of a genetic algorithm consider the problem of finding the fastest path for a truck that must visit 10 different camps that are connected by a complex network of roads. The genetic code could in this case be a list of camp names in the order that the truck visits the camps. The genetic search algorithm would generate a first generation of 40 random lists of camp names. A simulation module tests each list and finds the twenty lists that correspond to the shortest time for completing the visits. The next generations are constructed by

1. Copying the 20 best lists,
2. Mating of pairs of lists from the set of the 20 best so that each pair of parents produces two offspring, and
3. Random mutation of offspring genomes.

For sexual reproduction, we use the *order 1 crossover operator* which means that a group of camp codes first are copied from parent A to the child (grey area in the example below). The remaining camp codes in the child are selected in the order that they appear in the genome of parent B. This variant of sexual reproduction is adapted to the constraint of having precisely one of all camp codes in the child list.

Parent A: 1 4 3 7 6 2 8 5 9 0

Parent B: 4 5 6 7 8 9 0 1 2 3

Child : 4 5 3 7 6 2 8 9 0 1

Mutation means that two randomly selected camp codes are switched in some of the children after that they have been generated by order 1 crossover. After running this process for many generations, a good solution will be found. Because of the random nature of the process it is, however, not guaranteed that the globally optimal solution is generated.

4.3.2 Genetic search in program spaces

A computer program is the most expressive representation of solutions that is possible. Computer program languages that include a certain minimum set of language features belong to the set of *universal program languages*. A universal computer language can express all possible mechanical procedures that operate on digital data. A program that is expressed in a universal computer language can be translated to any other universal computer language without loss of functionality. All commonly used programming languages such as Fortran, Java, C, C++, etc. are universal. Searching for the solution in the space of computer programs written in a universal language should therefore be the most generic search process that we can imagine.

Genetic search is particularly suitable for exploring very rich and complex solution spaces and the program space is the richest and most complex space that we can think of. *Genetic programming* is the science of applying genetic search methods to program spaces. As a simple example of genetic programming, consider programs that can be expressed as trees where nodes are *operators* or *variables* and edges connect operators to the variables that they are operating on. An operator takes variables as inputs and outputs a set of variables. The equation $z=x+y$ can for example be understood as the + operator taking variables x and y as input and outputting the z variable. Figure 8 shows how the expression $z= x^2+x+y$ can be displayed as a tree. In the following we will describe genetic programming using the tree representation of computer programs.

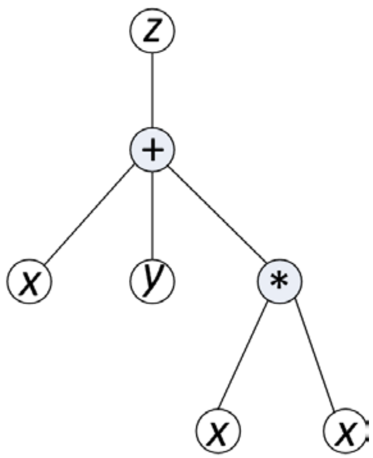

Figure 8. The program tree representation of the expression z= x^2+x+y.

Imagine now that we would like to have a computer program that predicts the fuel consumption *F* of a truck depending on the payload weight *M*, the driving distance *D*, the type of lubrication oil *L* used and the type of driving certificate *C* carried by the driver. Over the years, we have compiled a huge excel sheet of data where each row corresponds to one driving mission and consists of specific values of *F*, *M*, *D*, *L* and *C*. A genetic programming algorithm is assigned to automatically find a computer program that predicts fuel consumption depending on the four input variables.

The program uses the genetic search approach with sexual reproduction and mutation operators that are suitable for programs. Sexual reproduction of program trees is a generalization of the list form and corresponds to,

1) Copy the program of one of the parents and optionally remove a randomly selected sub-tree.
2) Cut out a sub-tree from the other parent and graft it to the program tree of the child at a randomly selected point.

Note that this means that a subroutine of one fairly successful program is implanted in another fairly successful program. It is easy to imagine that this sometimes could be beneficial. Figure 9 illustrates how sexual reproduction of fuel consumption programs may work.

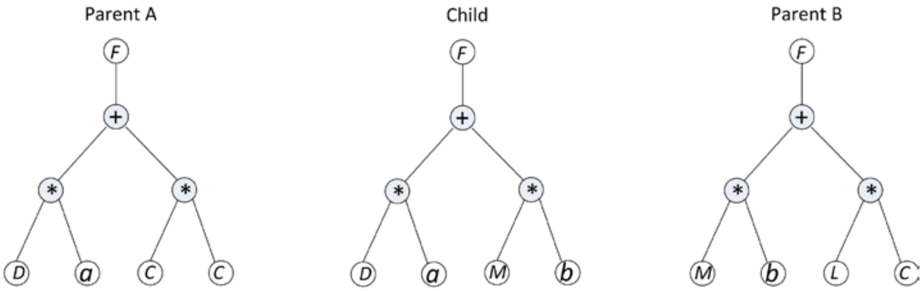

Figure 9. One step in the evolution of a fuel consumption estimator. Symbols *a* and **b** are constants. Parent A includes a sub-tree that indicates that fuel consumption is linearly dependent on driving distance D. Parent B has a sub-tree that connects fuel consumption to payload weight. Both parent also have sub-trees that don´t contribute to prediction accuracy. The child program that merges the successful sub-trees from both parents would be a successful program.

Mutation of programs means that the operator in one of the nodes is replaced and that the inputs of the operator are rearranged to satisfy the demands of the new operator. Mutations ensure that fundamentally new subroutines occasionally are produced. Each new generation is tested on the data. It is customary to favour not only data fidelity but also brevity. Short programs are thought to have better generalization ability while long programs tend to model statistical fluctuations in the data and thus have less predictive power. Running this process for many generations leads to more and more accurate programs that eventually will be able to model the training set accurately.

4.3.3 Manageable novelty by precise formulation of value

The default reference for this subsection is (Viriakis, 2003).

Viriakis (2003) applies GENETICA, a generic computer language for genetic programming, to the problem of architectural design. In a first step, GENETICA is used for developing an architectural design language called G-CAD where much of the domain knowledge is built in. In a second step, Viriakis applies G-CAD to the problem of designing a floor plan for a small hotel. The fitness function includes many constraints and one optimization goal. The constraints are precisely expressed in G-CAD and correspond to requirements including,

1) Eight identical apartments of a given size
2) A corridor with access to the apartments
3) A stairwell connecting the floor to other floors
4) Bathrooms, furniture, etc. in each apartment

The actual list of constraints is much longer and very precise. The optimization goal is to minimize the floor area while fulfilling the constraints. Due to the extensive and precise specification of the fitness function, G-CAD produces a fairly short list of design proposals each of which is suitable for implementation. Depending on the mindset and experience of the receiving architects some of the G-CAD proposals could be considered creative. The focus of G-CAD is to ensure the value of the output. Creativity is a secondary objective. Since GENETICA is a generic language it is quite feasible to build specialized daughter languages for many different domains. G-CAD is an inspiring success in the specific domain of architecture. The G-CAD approach of incorporating extensive domain knowledge and detailed constraints has the effect of giving human evaluators a short list of suggestions that is easy to evaluate and apply. The drawback is that some of the solutions that are filtered out by stringent application of constraints may be unfit for direct application but could serve to give humans creative insights. Breaking out of preconceived constraints is sometimes the key to success.

4.3.4 Balancing goal focus and associative fluidity

The default reference for this subsection is (DiPaola and Gabora, 2011).

Human creativity is a mix of goal oriented discipline and associative thinking outside the box. Discipline alone confines us within the boundaries of our traditions. Unbridled free association produces randomness but not value. Great painters tend to mostly adhere to a given style of painting but occasionally break out of stylistic constraints. DiPaola and Gabora (2011) use this insight to build an automated painter of abstract portraits.

The core of their method is a variant of genetic programming called Cartesian genetic programming (Miller and Thomson, 2000). Each node in the program corresponds to a pixel in the image. The inputs to a node are the pixel (x, y) coordinates; the outputs are the three hue, saturation and intensity parameters that describe the colour of the pixel in the HSV colour space.

The interesting contribution of DiPaola and Gabora (2011) is a new form of fitness function that evaluates members in the population of programs and decides which genetic code that will survive and reproduce. This fitness function is designed to balance goal focus and associative fluidity. Goal focus is attained by comparing each candidate painting to a target image. DiPaola and Gabora (2011) use a classical portrait of Charles Darwin as the target image. In the early phases of program evolution, target likeness dominates the fitness function. This would correspond to a human painting student that tries to make an exact copy of the original. As the evolutionary process proceeds to increasing similarity between the genetic programming population and the target image, the DiPaola and Gabora (2011) algorithm increases the influence of three artistic quality measures in the fitness function. The fitness function is hence a balanced mix of the target likeness measure and the artistic quality measures. The artistic measures are related to composition, tonality and colour theory respectively. The increasing influence of artistic measures corresponds to the experienced painter´s deliberate sacrifice of superficial likeness in favour of esthetical qualities in abstract portrait painting.

The work of DiPaola and Gabora demonstrates how genetic programming can be the foundation of computational creativity software and their introduction of a mix of goal orientation and creative exploration in the fitness function could inspire applications to other fields that also need a mix of discipline and fluidity. Starting with a high goal focus and gradually increasing the creative drive might be a good generic strategy for balancing novelty and value.

5 Conclusions

Creativity is a critical resource in many fields of human endeavour; in particular in fiercely competitive domains and in conflicts with intelligent adversaries. The main reason for the rising demand for creativity is the accelerating tempo, mounting complexity and increased level of intelligence in the modern world. Education, training and better organization boost human intelligence and the capacity of machine intelligence is growing at an exponential pace. Getting the right idea at the right time is vital for staying ahead of the competition and for avoiding to be outsmarted by intelligent antagonists.

Making maximum use of human creativity should be a chief concern of modern organizations. Computational creativity can in a near future provide an extra competitive edge when the stakes are high and could be vital in special cases. As the capacity of machine intelligence increases, the scope for computational creativity will widen. Creativity can be applied to strategic, operational and tactical levels of decision-making.

In very high-stake strategic decisions it is important to compile ideas from as many independent minds as possible in order to avoid costly mistakes caused by mental conformity and group-think. Computational creativity can provide a truly independent perspective. A computer using a transformative technique such as genetic programming is a very alien mind and could inspire ground-breaking insights.

Operational decision making in fast-paced conflicts needs creativity for getting inside the opponents decision loop. Decisive advantages can be won by more intelligent planning at a higher tempo than the competition can handle. Human creativity can be suppressed by stress and fatigue, which means that there is a need for creative support in operations. Computational creativity can aid for example by suggesting courses of action, identifying situation anomalies (Jändel 2013a) and by counterdeception planning (Jändel 2013b).

Tactical decision making progressively involves competition against smart machines that communicate at the speed of light and process information millions times faster than humans. Increasing machine intelligence means that more and more creativity is needed in tactical situations and that creative inputs sometimes are needed within microseconds. Only computational creativity can operate at machine speed. Even very simple forms of automated creativity such as pattern matching in plan libraries could be decisive in tactical situations.

The cyber arena is a prime example of how creativity is a key resource in modern conflicts. Strategic decision makers struggle with how to incorporate cyber aspects in high-level planning. Few leaders have experience of strategic cyber engagements. There are no doctrines to fall back on and it is even hard to find the right expertise to guide strategic decision makers in the sharply shifting cyber technologies. Getting the right ideas on the table before decisions of national importance are made is of paramount importance. Cyber operations depend on a large set of complex and interlocking technologies. This means that the range of options and possible plans is huge. Again, there is no field-tested handbook to rely on and being predictable is anyway disastrous in cyber operations. Creative operational planning is paramount in a cyber-world characterized by a dense web of deception and counterdeception. The tactical level of cyber conflicts is fought by intelligent software at machine speed. Creativity at this level must be automated to be effective at all.

The present research in computational creativity that has been reviewed in this report is a harbinger of greater things to come. It is not the state of the art of the branch of science that we presently call computational creativity that is important but that some of that research demonstrates the possibility of machine creativity. Computer science is increasingly concerned with higher levels of intelligence. A much more focused effort on automating aspects of creativity will be driven by demands from fields such as robotics, the semantic web, cyber warfare and not the least commercial tools for personal productivity and entertainment. Crucial near-time security applications of computational creativity will probably be in robotics and in the operational and tactical levels of cyber conflicts.

6 References

Aamodt, A. & Plaza, E. (1994), 'Case-Based Reasoning: Foundational Issues, Methodological Variations, and System Approaches', Artificial Intelligence Communications 7:1, 39-52.

Boden, M. (1992), The Creative Mind, Abacus, London.

Brandt, L. & Brandt, P. (2002), 'Making Sense of a Blend', Apparatur 4, 62-71.

Bushinsky, S. (2009), 'Deus Ex Machina—A Higher Creative Species in the Game of Chess', AI Magazine 30, 63-70.

Chatman, S. (1978), Story and discourse : narrative structure in fiction and film, Ithaca; London: Cornell University Press.

Chen, Z. (1998), 'Toward a better understanding of idea processors', Information and software technology 40, 541-553.

Chen, Z. (1988), 'COGMIR, a Computer Model for Knowledge Integration', PhD thesis, Louisiana State University, Baton Rouge.

Choi, J. W.; Joo, J. W. & Cho, D. L. (2002), Situation/Threat Assessment Fusion System (STAFS), in 'Fifth International Conference on Information Fusion', International Society of Information Fusion, ONERA-DTIM , BP 72, 29 Av. de la Division Leclerc, Chatillon, 92320, France, URL:http://www.inforfusion.org/], , pp. 1374-1380.

Di Paola, S. & Gabora, L. (2009), 'Incorporating characteristics of human creativity into an evolutionary art algorithm', Genetic Programming and Evolvable Machines 10(2).

Fauconnier, G. & Turner, M. (2002), The Way We Think: Conceptual Blending and the Mind's Hidden Complexities, Basic Books.

Franke, U.; Johansson, R.; Mårtenson, C. & P., S. (2012), 'Fusion av information från heterogena källor.', Technical report, Swedish Defence Research Agency, FOI-R–3453–SE.

Gervás, P. & León, C. (2010), Story Generation Driven by System-Modified Evaluation Validated by Human Judges, in 'Proc. of the International Conference on Computational Creativity'.

Gervás, P.; Pérez y Pérez, R.; Sosa, R. & Lemaitre, C. (2007), On the Fly Collaborative Story-Telling: Revising Contributions to Match a Shared Partial Story Line, in 'Proc. of International Joint Workshop on Computational Creativity'.

Giora, R. (2003), On our Mind: Sailience, Context and Figurative Language, Oxford University Press.

Griva, I.; Nash, S. & Sofer, A. (2008), Linear and Nonlinear Optimization (2. ed.), SIAM.

Hassan, S.; Gervás, P.; León, C. & Hervás, R. (2007), A Computer Model that Generates Biography-like Narratives, in 'Proc. of International Joint Workshop on Computational Creativity'.

Jändel, M. (2013a), 'Computational Creativity in Naturalistic Decision-Making', Submitted to International conference on computational creativity, 2013.

Jändel, M. (2013b), 'Computational Creativity for Counterdeception in Information Fusion', Submitted to the 16th International Conference on Information Fusion.

Jändel, M. (2012), 'Maskinkreativitet för beslutstöd'(FOI memo 3974), Technical report, Swedish Defence Research Agency, In Swedish.

Klein, G. (2008), 'Naturalistic Decision Making', Human Factors 50, 456-460.

Klein, G. (2003), Intuition at work, Doubleday, New York.

Klein, G.; Calderwood, R. & Clinton-Cirocco, A. (1986), Rapid decision-making on the fireground, in 'Proceedings of the Human Factors and Ergometrics Society', pp. 576-580.

Koestler, A. (1964), The Act of Creation, Macmillan.

Lee, W. & Xiang, D. (2001), Information-theoretic measures for anomaly detection, in 'Proceedings of the IEEE Symposium on Security and Privacy'.

León, C. & Gervás, P. (2008), Creative Storytelling Based on Transformation of Generation Rules, in 'Proc. 5th International Joint Workshop on Computational Creativity'.

Li, B. & Riedl, M. O. (2011b), Creative Gadget Design in Fictions: Generalized Planning in Analogical Spaces, in '8th ACM Conference of Cognition and Creativity'.

Li, B. & Riedl, M. O. (2011a), A Phone That Cures Your Flu: Generating Imaginary Gadgets in Fictions with Planning and Analogies, in '4th Workshop of Intelligent Narrative Technologies'.

Li, B.; Zook, A.; Davis, N. & Riedl, M. .. (2012), Goal-Driven Conceptual Blending: A Computational Approach for Creativity., in 'Proceedings of the 2012 International Conference on Computational Creativity'.

McCaffrey, T. & Spector, L. (2011), How the Obscure Features Hypothesis Leads to Innovation Assistant Software, in 'Proceedings of the Second International Conference on Computational Creativity'.

McCormack, J. & d'Inverno (eds), M. (2012), Computers and Creativity, Springer-Verlag, Berlin Heidelberg.

McGregor, S. (2007), Algorithmic Information Theory and Novelty Generation, in 'International Conference on Computational Creativity', pp. 107-112.

Meehan, J. R. (1977), TALE-SPIN, An Interactive Program that Writes Stories, in 'Proc. of the Fifth International Joint Conference on Artificial Intelligence'.

Miller, J. & Thomson, P. (2000), Cartesian genetic programming, in 'Proceedings of the 3rd European conference on genetic programming', pp. 121-132.

Multiple unidentified authors (2011), 'Creativity and ITC: Future and Emerging Technologies Consultation Workshop', Technical report, European Commission.

Pavón, J.; Arroyo, M.; Hassan, S. & Sansores, C. (2006), Simulación de sistemas sociales con agentes software, in 'Actas del Campus Multidisciplinar en Percepción e Inteligencia, vol. I'.

Peinado, F.; Gervás, P. & Moreno-Ger, P. (2005), Interactive Storytelling in Educational Environments, in 'Proc. of the 3rd International Conference on Multimedia and ICT´s in Education: Recent Research Developments in Learning Technologies, III'.

Perez y Perez, R.; Negrete, S.; Peñaloza, E.; Castellanos, V.; Ávila, R. & Lemaitre, C. (2010), MEXICA-Impro: A Computational Model for Narrative Improvisation, in 'Proc. of the International Conference on Computational Creativity'.

Rank, S.; Hoffmann, S.; Struck, H.-G.; Spierling, U. & Petta, P. (2012), Creativity in Configuring Affective Agents for Interactive Storytelling, in 'Proc. of the 3rd International Conference on Computational Creativity'.

Riedl, M. & Young, M. (2005), Story planning as exploratory creativity: Techniques for expanding the narrative search space, in 'Proc. of 2005 IJCAI workshop on computational creativity'.

Riedl, M. O. (2008), Vignette-Based Story Planning: Creativity Through Exploration and Retrieval, in 'Proc. 5th International Joint Workshop on Computational Creativity'.

Schmidhuber, J. (2006), 'Developmental Robotics, Optimal Artificial Curiosity, Creativity, Music, and the Fine Arts', Connection Science 18, 173-187.

Simon, M. (2007), Algorithmic Information Theory and Novelty Generation, in 'Proceedings of the 4th International Joint Workshop on Computational Creativity', pp. 109--112.

Sims, K. (1991), Artificial Evolution for Computer Graphics, in 'SIGGRAPH '91 Proceedings', pp. 319-328.

Smith, A. & Mateas, M. (2011), Knowledge-Level Creativity in Game Design, in 'Proc. of the 2nd International Conference on Computational Creativity'.

Strapparava, C.; Valitutti, A. & Stock, O. (2007), Automatizing Two Creative Functions for Advertising, in 'International Conference on Computational Creativity'.

Surdu, J. R. & Kittka, K. (2008), The Deep Green concept, in 'Proceedings of the 2008 Spring simulation multiconference', Society for Computer Simulation International, San Diego, CA, USA, pp. 623--631.

Swartjes, I. & Vromen, J. (2007), Narrative Inspiration: Using Case Based Problem Solving to Support Emergent Story Generation, in '4th International Joint Workshop on Computational Creativity'.

Tan, K.-M. T. & Kwok, K. (2009), Scenario Generation Using Double-Scope Blending, in 'AAAI Fall Symposium'.

Theune, M.; Rensen, S.; op den Akker, R.; Heylen, D. & Nijholt, A. (2004), Emotional characters for automatic plot generation, in 'Technologies for Interactive Digital Storytelling and Entertainment (TIDSE)'.

Viriakis, L. (2003), 'GENETICA: A computer language that supports general formal expression with evolving data structures', IEEE Transactions on evolutionary computation 7(5), 456-480.

Zhu, J. (2012), Towards a New Evaluation Approach in Computational Narrative Systems, in 'Proc. of the 3rd International Conference on Computational Creativity'.

Zhu, J. & Ontañón, S. (2010), Towards Analogy-Based Story Generation, in 'Proc. of the International Conference on Computational Creativity'.

FOI, Swedish Defence Research Agency, is a mainly assignment-funded agency under the Ministry of Defence. The core activities are research, method and technology development, as well as studies conducted in the interests of Swedish defence and the safety and security of society. The organisation employs approximately 1000 personnel of whom about 800 are scientists. This makes FOI Sweden's largest research institute. FOI gives its customers access to leading-edge expertise in a large number of fields such as security policy studies, defence and security related analyses, the assessment of various types of threat, systems for control and management of crises, protection against and management of hazardous substances, IT security and the potential offered by new sensors.

FOI
Swedish Defence Research Agency
SE-164 90 Stockholm

Phone: +46 8 555 030 00
Fax: +46 8 555 031 00

www.foi.se