

CREATIVITY, REQUIREMENTS AND PERSPECTIVES

Oliver Hoffmann¹⁰, David Cropley¹¹},¹², Arthur Cropley¹³
Lemai Nguyen¹⁴, and Paul Swatman¹⁵

ABSTRACT

Is there room for more creativity in information systems? This article grew out of an AWRE'04 panel discussion on creativity in requirements engineering, and the impact of requirements engineering on creativity in systems engineering and systems use. Both panel and article were motivated by the goal of identifying a framework for understanding creativity in a larger context and thus establishing a potential structure for future research. The authors' research backgrounds differ widely and, at times, our views conflict – occasionally, quite sharply. We make underlying world views – our own and those of relevant disciplines – explicit; identify the paradox caused by the need to be functionally creative while leaving room for creativity in successive stages; and argue for a multi-paradigm framework for resolving this paradox.

Keywords: information systems, creativity, requirements engineering, world views

INTRODUCTION

Creativity is widely acknowledged as vital to society. Higgins (1994) lists ten “*primary challenges*” that business faces in the twenty-first century, and suggests that innovation and creativity are the mechanisms to address these challenges. There is also an important link between creativity and social welfare. Disciplines such as business, industry, trade and commerce, international relations, science, medicine, engineering and, of course, information systems all suggest responses to those challenges. The link between the challenge and these disciplines is not simply a matter of social or educational philosophy: The importance of creativity to prosperity, even economic survival, is stressed by many authors. “*The economic importance of creativity is clear because new products or services create jobs*” (Sternberg & Lubart 1992). To take a particularly striking example of the concrete importance of creativity, the present era of global terrorism requires that technological solutions to the problems posed by terrorist activities exhibit high levels of creativity, if they are to withstand the efforts of terrorists to circumvent them. How do information systems and creativity relate? According to the United States National Academies Committee on Information Technology and Creativity (2004), the effects would be mutually supportive: “*Just as the engagement of IT helps shape the development of inventive and creative practices, so also can inventive and creative practices positively*

¹⁰ oliver@hoffmann.org

¹¹ David.Cropley@unisa.edu.au

¹² Systems Engineering and Evaluation Centre, University of South Australia

¹³ acropley@bigpond.net.au, Emeritus Professor of Educational Psychology, Hamburg University

¹⁴ lem.ai.nguyen@deakin.edu.au, School of Information Systems, Deakin University

¹⁵ Paul.Swatman@unisa.edu.au, Computer and Information Science School, University of South Australia

influence the development of IT". The Committee on Information Technology and

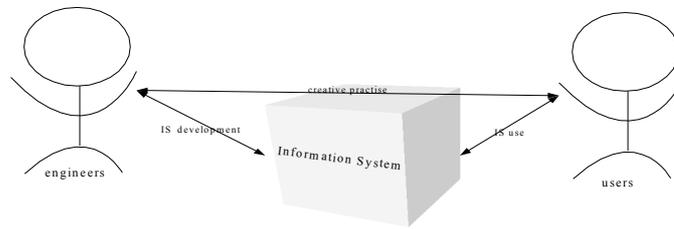


Figure 13: IS and creative practise: potential mutual benefits

Creativity (2004) refers to “*creative practices*”, primarily arts related, but the claim of mutually beneficial interaction between information technology and creativity is made in a quite general way. Does the increased use of information systems enhance human creativity? Does increased engineering creativity improve the quality of information systems? What do we have to know about information systems in order to make more room for creativity?

Although the topic of creativity would, in general, include “*creative information systems*”, where the system itself displays creative behaviour, we will exclude this aspect from our article and focus, instead, on human creativity in relation to information systems. Roughly speaking, there are two actor-sets involved: those developing information systems (whom we might term “*engineers*”) and those using them. We will introduce a distinction within the former category, dividing the group of engineers into requirements engineers and system designers. Even though these three groups do not necessarily interact with each other directly, they are all linked by their involvement with information systems, and the actions of each influence the creative potential of all.

We will analyse what is known about the role of creativity in these three loci, and how differences in their understandings and conceptualisations might inform a more general framework. We will start our analysis with the last group: What expectations do creative users have of information system properties?

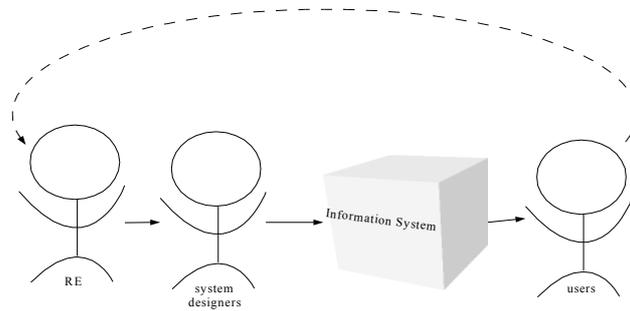


Figure 14: actor-sets linked by their involvement with IS: three loci for human creativity

WHAT CREATIVE USERS EXPECT

Supporting creativity is not necessarily the same as increasing productivity. Rather than simply making our routine activities more efficient (essentially a *quantitative* change), information technology can achieve *qualitative* effects by extending creative potential, thus helping to “*reframe and redirect the expenditure of human effort, generating unanticipated payoffs of exceptionally high value*” (Committee on Information Technology and Creativity 2004). On the other hand, the use of computing can also decrease creative potential: computer software always relies on specific metaphors, and enforcing the wrong metaphor can inhibit users from exploring creative alternatives, leading to solutions that are rich in detail but lack core creativity (Lawson 1999). As with all tools, the use of information systems introduces some new capabilities, while restricting others. But information systems are not like other tools. Being programmable and adaptable, information systems should be less likely to impose a particular way of working than other tools. To creative users, information systems might not seem like tools at all, but rather like instruments (Beynon et al. 2001): they present obvious benefits and limitations, both of which must be studied and mastered, and which can both be used in creative ways. Eventually, creative masters can achieve a form of harmony or flow (Csikszentmihalyi 1997) when interacting with their instrument.

But are information systems built with such a use in mind? As we will discuss below, increasing information systems development creativity might actually decrease potential creative use: information systems engineers display their creativity via the power of their creations, and typically, more powerful computing is regarded as equivalent to empowering computer users. For creative information systems use, however, this might not always be the case.

Any information systems user can be creative. But for some, like artists, creative designers, scientists developing a new theory, or engineers working on innovative solutions, creativity is an essential goal. Like a magnifying glass, these groups of users expose the role of information systems in creative processes. So what do creative users want information

systems to do for them? In some ways, they expect the same basic services as all users: computation, storage, and communication. “*Tasks for helping people be more creative more of the time*” Shneiderman (2002) can be classified according to the services they depend on (Shneiderman’s tasks in boldface):

IS service for creative users 1. *modelling.*

The ability to compute on existing information can serve as the basis for various kinds of models, which can be used for **visualisations** of objects and processes, **exploration** of scenarios and of complete works from various fragments.

IS service for creative users 2. *book keeping.*

Not surprisingly, the ability to store information is one of the key reasons to use an information system, including for creative purposes. This service helps with tasks such as **reviewing** previous steps in a process and **searching** for compatible material.

IS service for creative users 3. *communication.*

Human creativity frequently (some argue always) involves some sort of collaborative effort, and IS can provide means for **relating** to others or **disseminating** results. Some creative computer users are also computer experts, but most are not. Subsequently, most see the computer as a *black box*, and are not interested in intricate system performance details, only in overall system behaviour towards them. From this perspective, behaviourist criteria for the needs of creative users can be formulated, for instance those compiled by Candy & Edmonds: *The system must enable the user to*

IS behaviour criterion 1. *adopt an holistic perspective on the task or problem*

To achieve this, the user should have access, for example, to overviews, multiple views and alternative representation of the data in respect of the developing solution.

IS behaviour criterion 2. *keep several channels of exploration open in parallel*

To achieve this, the user should be able to alternate between tasks freely, fluently and quickly. This implies having simultaneous access to different forms of visual data and methods for knowledge interaction.

IS behaviour criterion 3. *explore and evaluate existing design knowledge in relation to other heterogeneous knowledge sources, generate and evaluate new concepts and apply constraints as appropriate.*

Candy & Edmonds derive these criteria from studies of what they call *creative knowledge work*, primarily scientists’ use of information systems for analysis and theory development. Hewett (2005) uses the term *creative problem solving environments* for information systems supporting such work. Similar observations have been made in studies on artists (Edmonds & Candy 2002) and their use of information systems: While it is important that information systems help with modelling (IS service for creative users 1), it is equally important that the choice of model and representation is always left to the user, enabling a holistic overview (IS behaviour criterion 1). In other words, information systems

computing power has to be balanced with user control. With every new information systems capability, ways of maintaining user capability for informed choice has to be provided. Clearly, creative users expect information systems engineers to provide systems with such properties. Whether engineers are able to fulfil this expectation will in part depend on their understanding of creativity - that of information systems users and their own.

CREATIVITY IN ENGINEERING

According to Horenstein (2002), engineering is “*a career full of discovery, creativity and excitement*”. A simple definition of engineering is that it harnesses both natural resources and the laws/forces of nature to provide society with *products, systems and services*, for the good of humankind.

But what is engineering creativity? Burghardt (1995) put it plainly: “*technology is the manifestation of engineering creativity*”. The products of engineering creativity are physical objects, complex systems such as a submarine or a business information system, or processes in the sense of a service, technique or method (a manufacturing process, a control process, a logistics service).

What makes products creative? It seems more or less self-evident that the first characteristic of creativity is *novelty*—as the psychologist Bruner (1962) put it, creativity must create *surprise*. However, surprisingness is not sufficient on its own. If it were, every absurd suggestion would be creative. Thus, creative solutions must be not only novel, but also *relevant* and *effective* (Bruner 1962).

In artistic creativity or any form of *non-functional* creativity, surprisingness alone may be regarded as automatically being effective, or effectiveness may be judged according to purely aesthetic or expressive criteria. In the case of engineers, however, effectiveness means, in the first instance: “*Will it work?*”. Other *functional* criteria of effectiveness are whether or not a solution can be financed, whether it can be produced with available resources of skills and money, whether it is consistent with a firm’s culture, whether it will be accepted by customers or the public, and so on. If it will not, no amount of novelty, even novelty that does the job, can be inserted into what already exists, without great risk.

The consequences of generating novelty that lacks effectiveness are quite different for information systems engineers and information systems users. For an artist or author, for instance, it might mean embarrassment, damage to the reputation, or loss of income, although it might also mean praise for being *avant garde*. For engineers in general, however, it might well mean bankruptcy for a firm, endless lawsuits, or even a disaster causing great loss of life, such as the collapse of the West Gate Bridge on October 15, 1970, Australia’s worst industrial accident. Thus, paradoxically, in engineering creativity *effectiveness takes precedence over novelty*, as absurd as that may seem. Novelty without effectiveness is simply too dangerous!

This means that for engineering, there is a tension between the need to produce novelty and the necessity of effectiveness, with the latter likely to be emphasized at the cost of the former. Cropley & Cropley (2005) refer to creativity possessing this particular property as *functional* creativity, where the emphasis is not only on characteristics such as novelty, but also on the important engineering concept of *effectiveness*, forming “*...creativity with a purpose*” (Burghardt 1995).

This is not to say that non-engineering creativity is devoid of purpose: As mentioned above, creativity in general is essential for progress and growth, which in itself is enough to give

all creative efforts purpose. But in contrast to engineering creativity, general creativity cannot be judged against predefined criteria of relevance or effectivity. To enable creative users to achieve a more *holistic* view - as they desire - information systems engineers would have to have a clear understanding of the differences between their own creative process and the needs of their clients. But as we will show, the tension between engineering precision and human creativity extends to the process of formulating such needs as well.

CREATIVITY IN REQUIREMENTS ENGINEERING

Engineering is focused on meeting the needs of users or customers - humans or humankind (Volland 2004). From a pure *engineering* point of view, the purpose of Requirements Engineering (RE) is to translate that need (often ill-defined and vague) into a formal, rigorous statement, so that engineers can build a satisfactory solution to it.

When aiming at translating creative user needs for information systems designers, one would therefore expect RE to do the job. But as in the general engineering case, this runs contrary to how requirements engineers see their own work. Like other kinds of engineering, RE is typically seen as a problem solving process. "*RE can be defined as the systematic process of developing requirements through an iterative co-operative process of analysing a problem, documenting the resulting observations in a variety of representation formats, and checking the accuracy of the understanding gained.*" (Loucopoulos & Karakostas 1995).

Based on this definition, RE problem solving is described as an application of engineering methods and techniques to analyse and solve a requirements problem. A large proportion of RE research efforts has been placed on the development and evaluation of various engineering modelling methods and techniques to express and specify requirements more completely and precisely.

What place does creativity have in RE? Many RE authors and especially RE practitioners acknowledge and agree on the importance and the role of creative techniques and heuristics in applying proposed engineering principles to solving requirements problems. But how creativity occurs in this engineering application has somehow been neglected in RE research. How should requirements engineers incorporate the needs of creative users if their own creativity is hardly accommodated? Requirements engineers' creativity enables them to deal with new problems/domains, unique contexts, or novel applications and combinations of existing methods and techniques. These abilities are assumed to be based either on "*mysterious*" natural intuitions, heuristics which accumulate over years of experiences, and/or a deliberate application of creativity techniques such as brainstorming, mind mapping, scenario simulation, and so forth (Gause & Weinberg 1989; Batra & Davis 1992; Shanks & Simsion 1992).

Creativity in RE has recently received increasing attention in the RE community. Nguyen et al. (2000) reported that requirements engineers occasionally re-conceptualise the users' requirements and restructured their requirements model as a result of unexpected insight. Maiden & Gizikis (2001) pointed out the lack of and calls for more research into supporting creativity in RE. Since then creativity workshops and tutorials have been continually held at AWRE'02, RE'03, RE'04, AWRE'04 and RE'05 conferences. Recent research effort into creativity in RE can be classified into the following main streams: understanding and facilitating the creative problem solving process (Nguyen & Swatman 2003; Pennel & Maiden 2003; Maiden et al. 2004; Nguyen & Swatman 2006), developing brainstorming techniques to undertake requirements elicitation (Mich et al. 2004), incorporating creativity

in RE education (Armarego 2004), and investigating context factors and socio-organisational context within which RE creativity is exercised (Cybulski et al. 2003; Dallman et al. 2005). In our view, to effectively foster and support creativity in RE it is vital to understand how the RE process occurs - to define the creative path from requirement problem to solution.

Action research conducted by Nguyen & Swatman (2000, 2003) has yielded some unexpected findings on the creative path from requirement problem to solution: Viewed from the *problem solving* perspective, creative steps appear as catastrophic interruptions. Consequently, the RE problem solving process can be described as a catastrophe-cycle process consisting of cycles of building up an understanding and structuring of the problem and occasionally re-conceptualising and restructuring it through opportunistic gaining of accidental knowledge about the problem. This process can be explained through an evolution of three types of complexity in the requirements model: essential, incidental and accidental complexity.

Essential complexity represents the intrinsic understanding of the requirements problem gained, agreed upon and embedded in the requirements model. Ideally, this type of complexity grows over time towards an objective *completeness* of the problem complexity. Essential complexity is rather subjective and may be assessed as agreed knowledge about the real world problem between different stakeholders. Overall, we can assume that our understanding of the problem must not be decreasing as the problem space is explored and new requirements are uncovered by the requirements engineer. Therefore, essential complexity grows over time as learning takes place, the problem space is explored, and different creative ideas and thoughts about user needs are elicited, discovered, invented, expressed and added into the requirements model.

Incidental complexity represents complexity of representation rather than substance in the model. It shows the poor fit between the structure of the requirements model and the need that the model attempts to mirror. As new essential understanding is continuously gained and agreed upon, new requirements are added to the model. The more requirement components are added into the model, the more difficult it becomes to add new components into the increasingly complex existing model. Therefore, incidental complexity grows exponentially as the structure of the model grows.

As new model components are expressed, added and modified in the requirements model, incidental complexity increases due to inevitable entropy and a growing gap between the requirements model and the real world need. Therefore, the overall complexity grows consisting of both essential understanding gained, and incidental complexity growth. During this process, creativity plays a role in requirements expressions and in modelling and evaluation of different modelling choices using various criteria (for example see Shanks & Simsion 1992). However, the most significant creative steps occur when the entire requirements model is transformed.

At some unexpected "*Aha!*" moment, the requirements engineer re-conceptualises the problem, and restructures the requirements model based on the new essential knowledge formed. From the classical *problem solving* perspective, this moment appears as a disastrous disruption. Accidentally discovered essential knowledge can be referred to as *accidental complexity* to denote a significant change in the model structure which now reflects a new essential understanding of the real world problem. Accidental complexity represents hidden knowledge in the requirements model which becomes explicit only as a result of re-conceptualisation insight at the crisis points. After the model is restructured, the accidental complexity becomes a part of the essential complexity which continues to grow

over time. As a result of the re-conceptualisation insight, the model is restructured and the essential complexity increases in the form of accidental complexity while incidental complexity decreases. The newly restructured model becomes the new basis for the next catastrophe cycle of the construction and restructuring of the requirements model.

This type of engineering creativity disguised as disastrous disruption was discovered while attempting to uncover cognitive behaviours of requirements engineers. A number of authors (such as Guindon 1990; Visser 1990; Khushalani 1997; Carroll & Swatman 1999; Nguyen et al. 2000) suggest that the RE process be understood as an insight-driven and opportunistic rather than a systematic, top-down and balanced development process.

However, how to trigger insight or make restructuring happen seem to remain an unanswered question: *“A restructuring event has an involuntary character; it is experienced as something that happens, rather than as something the problem solver does”* (Ohlsson 1984).

This may explain why insight and support for insight have received limited attention in RE. The evaluation stage therefore is either missing or not treated explicitly in the literature. Recently, Maiden and his colleagues (2004) utilised creative workshops to simulate and structure the RE creative process on the basis of Wallas’s (1926) model discussed below.

Nguyen & Swatman (2003, 2006) associate the preparation and incubation stages with the creative and reflective structuring of the requirements model. They suggest that various creativity techniques, such as brainstorming, idea generation, scenarios and simulations, and ad hoc design rationale could be used to support and promote creativity during these stages.

COGNITIVE PROCESSES

The above explanation of the RE catastrophe cycle process is consistent with Wallas’s (1926) description of the creative process. According to him, the creative problem solving process consists of four stages: preparation, incubation, illumination (insight), and the verification of illumination. Hadamard (1954) added that creativity involves an unconscious mental process and insight is seen as a breakthrough by unconscious ideas. The growth of essential and incidental complexity of the requirements model can be seen as a result of the preparation and incubation stages.

The insight (or *Aha!* moment) and its restructuring role in problem solving are often discussed and recognised in Gestalt psychology. The insight concept is based on the illumination stage in Wallas’s creative process model. Restructuring is often associated with the occurrence of insight, a sudden, unpredictable flash of ideas which often involves a surprise and a solution.

According to Gestalt psychologists, problem space restructuring is crucial in problem solving and reveals a new way of looking at the problem, often from a broader perspective. Where does novelty come from? Two concepts are very useful here:

- Non-algorithmic thinking: The first modern psychological analysis of creativity (Guilford 1950) laid down the basis for understanding creative thinking. He distinguished between *convergent* thinking, aimed at finding the one and one only best or correct answer through the application of conventional logic to existing knowledge (for instance: *“What number comes next in the series 2, 4, 8, ...”*), and *divergent* thinking, aimed at producing a variety of alternative answers to open problems (for instance: *“Give as many uses as you can for a tin can”*). The essential idea is that

of a contrast between thinking that follows “*the rules*” of logic, good sense, order, and so on, and thinking that breaks away from the usual patterns by going beyond the obvious, abandoning the known and branching out, crossing boundaries, making unexpected combinations, seeing remote associations, building networks of ideas, and so on. Resnick (1987) used the term *non-algorithmic* thinking to refer to the latter. However, psychological research has differentiated the simple equation: *divergent thinking = creativity*, and current research has re-established the value of *convergent* thinking in creativity (e.g. (Cropley 2006)).

- Nuanced judgement: Subsequently, psychologists have analysed creative thinking in terms not just of the kind of thinking (see the examples just given), but also in terms of *metacognitive processes*, i.e., the “*executive processes*” in thinking, that allow people to organize and keep track of their own non-algorithmic thinking. Metacognition may be defined as awareness of one’s own cognitive processes rather than the content of those processes together with the use of that self awareness in controlling and improving cognitive processes (Biggs & Moore 1993). Flavell, one of the first to introduce the concept of metacognition, suggests that metacognitive knowledge (knowledge about one’s own cognition) includes three interacting variables: knowledge of strategies, of person (self and others) and of tasks. Metacognitive processes include reviewing one’s own knowledge, consciously directing attention, identifying and marshalling one’s own cognitive resources, selecting a promising line of attack, designing a workable solution strategy, evaluating one’s own progress, recognizing and avoiding dead ends, recognizing promising alternatives, being able to make an effective change of course when it is needed, and so on. Such control processes involve what Resnick called *nuanced judgement*. This requires controlling thinking not just on the basis of logic and correctness/incorrectness, but also of hunches, personal inclinations, subjective decisions about probability, intuition, and so on.

16

Non-algorithmic thinking and nuanced judgement are facilitated by an open definition of the task, minimal specification of the solution pathway, and the loosest practicable definition of the solution. Precision, in as much as it constrains choice, is, in simplistic terms, anti-creative. To enable creative solutions, the engineer would have to tolerate living with *uncertainty, ambiguity, openness*, and show personal *flexibility*.

Unfortunately, such conditions are difficult for both engineers and clients to tolerate, because they expose both partners to *risk*, personal risk (ridicule, etc.), vocational risk (loss of the job), financial risk, and so on! Thus engineers typically dislike ambiguity, at a personal, cognitive level, so that the needs of creative engineering are at odds with the psychological profile of engineers. It is not hard to see why this situation might have arisen. Aside from the pressures of those who occupy the space later in the life cycle of systems

16 It is extremely important to note that nuanced judgement does not mean *no* judgement, or do whatever you please. It involves rigour, self-criticism, search for excellence, etc..., but on the basis of a different way of thinking.

(the designer, builder etc.), requirements engineers are likely to feel more comfortable reducing the ambiguity of a need by jumping in to constrain the solution. This seems to fit the general psychological profile of engineers.

EDUCATIONAL IMPLICATIONS

Fortunately, non-algorithmic thinking and nuanced judgement can be taught and learned in university classrooms. The same is true of risk acceptance and tolerance of uncertainty. What is needed is

- systematic thematisation of the issues in terms of understandable concepts,
- repeated exposure of students to uncertainty, and
- assessment based on rewards for dealing productively with ambiguity, taking risks and the like.

Cropley & Cropley (2000) give a specific example of how to do this in an engineering class. Encouraging the development of creative traits in students is challenging for teachers. When creative behaviour leads to students being less easy to manage, teachers tend to perceive them less as creative than as a nuisance: Teachers' rating of their favourite students is negatively correlated with creativity (Edwards 2001). But the resistance against non-algorithmic thinking goes both ways, and is a long-standing problem, as the following examples illustrates.

In 1903 there was an unusually high failure rate among Western Australian students who took the University of Adelaide's matriculation examination, and this led to an intense public discussion, for instance in the Morning Herald (1904). The complaint was straightforward: The university had suddenly set assignments such as "*To what extent can Sir Walter Scott's novels be classified as historically accurate?*", whereas what the parents and protesting teachers wanted was "*List three examples of historical errors in the novel Ivanhoe*". The complainants specifically criticized the fact that it would not be possible to learn the answer to the "*To what extent ...*" example by heart, and that students would have to use their own judgement, with all the risks associated with this, including the possibility of being wrong! Protesting parents and educators made it plain that they were not demanding an easy examination, but one that they saw as predictable, objective and fair.

As the former anecdote refers to students of the humanities, one can imagine how strong resistance to non-algorithmic thinking is in engineering, in students, teachers and practising engineers alike. Where too much algorithmic thinking can lead is demonstrated by an example in scientific rigour on part of a German-Latvian microbiologist and pathologist.

Eugen Semmer worked in the Institute of Veterinary Medicine in Riga. In 1870 he published a paper reporting on the strange recovery of two horses that were dying from what we would now call "*infections*". At the Veterinary Clinic the horses were accidentally exposed to spores of the mushroom *penicillium notatum* and inexplicably got better, despite a grave prognosis. Semmer correctly identified the mushrooms as the cause of the horses' recovery, but saw this as a problem that impeded his research on the pathology of disease: How can you conduct microbiological studies in pathology if the patients regain their health? Neither Semmer himself nor the readers of his paper recognized that he had, in fact, discovered antibiotics, presumably because this had not been part of the project specifications! The world had to wait another 70 years for Alexander Fleming to discover penicillin.

The tension between reliable functionality and creative potential is not unique to information systems engineering, but the players involved - requirements engineers, information systems designers and information systems users - are particularly heterogeneous and to a large degree isolated from each other. In fact the only thing that connects them is technology - the information systems they all refer to - and therefore the ways they influence each other might easily go unnoticed.

THE CREATIVITY PARADOX

The dimensions of a dilemma emerge: The more engineers do their job as traditionally expected, the less room remains for creativity. This dilemma manifests itself in the understanding of RE creativity, the relationship of RE with information systems design, and of information systems design and implementation with information systems use. Good requirements practice dictates that we should avoid specifying the solution prematurely (i.e. we should start by saying *what* the system must do, and delay specifying *how* it should do it for as long as possible). The requirements, in other words, should be *solution-free*. But effectiveness takes precedence over novelty, and engineers know that eventually someone has to cut metal (or code), which requires clarity, precision and money.

Thus requirements engineers seek to describe the task precisely, specify the pathway to the solution, and define in advance what will be an acceptable solution. In an effort to facilitate the next phases of the engineering process (design and synthesis), requirements engineers are under pressure to be too helpful to the designers. They are over-applying their own creative skills. Too many non-functional requirements (constraints) are introduced in the requirements process, to the point where all effective choice is removed, and subsequent creativity is restricted. The same paradox applies to information systems designers: While trying to empower information systems users with increased *functionality*, they might actually decrease the potential for *holistic* use of information systems, and thereby restrict information systems user creativity.

While seeking to promote creativity, requirements and their implementation may, because of the justified and necessary emphasis on effectiveness, act as a gatekeeper, holding creativity at bay. The paradox involves requirements engineers, engineers in general, and end users for engineered products, and it effects the creativity of all three groups. Thus, we call it the *creativity paradox*: Over-applying the *problem solving* notion may inhibit requirement engineers creativity, over-applying requirements engineering may inhibit system design creativity, and over-applying engineering may inhibit product user creativity. Resolving a paradox typically involves introducing a new frame of reference. In this case, we start by taking a closer look at the notion of *problem solving* used in information systems engineering. Throughout RE and SE literature, problems in RE are often described as *ill structured*, *complex*, and *domain specific* (Guindon 1990; Batra & Davis 1992). The *ill structuredness* of RE problems lies in the ambiguity, vagueness and incompleteness of human wants and needs and the possibility of multiple solutions. The complexity in RE manifests itself in different levels of abstraction and varying aspects of requirements (data, process, object, method, etc.) and various formats of representation ranging from informal, semi-formal to formal. In addition, requirements are rather volatile and evolve during the project partly due to our exploration in the problem space, partly due to the dynamics of a business environment formed and reformed by the interactions of the stakeholders. Thus, this kind of problems in RE makes *sense* only in a specific domain and often unique context. Interestingly, solutions to such ill-structured, complex, and domain specific

problems are expected to be well defined, precisely expressed and completely specified, so that they will serve as input to systems design and the implementation phase.

Considering all these properties, is *problem solving* the right metaphor for describing this activity? On the one hand, Coyne (2005) argues that in fact all problems are *wicked*, or ill-defined. On the other, the fundamental usefulness of the problem-solving metaphor is placed in question. Robertson for instance suggests that requirements engineers should be creative and even invent requirements based on their understanding of the organisation's competitive business goals and context.

Due to subjectivity in problem perception, formulation, judgement and acceptance of suggested solutions, a clear understanding of insight is important. None of the original elements for the classic definition of problem solving can be upheld: Neither the problem space is clearly defined or stable, nor are there objective criteria for testing solutions or stating the problem itself. In addition, problem solvers themselves are significantly changed during the problem solving process, by learning and therefore transforming their own perception.

The path from a problem to a possible solution in RE is created and crafted by requirements engineers when they explore the problem space from subjective descriptions of the problem by different stakeholders. This involves continuously going backwards and forwards between different expressions of needs and wants to develop a problem space as agreed by the problem owner(s) and solver(s). The problem solver plays an active and creative role in shaping and crafting the problem through consulting and helping the problem owner(s) understand and express their needs and possible requirements (Gause & Weinberg 1989; Maiden et al. 2004). This is a learning process during which both problem owner(s) and solver(s) learn about the problem.

It is expected that RE solutions will make sense in the systems world (Checkland & Scholes 1990) and will be expressed in the form of a requirements model/specification. The solution (in the systems world) has to mirror the need (in the real world). Consequently, the assessment of solutions is rather subjective and contextual. Pohl (1994) refers to this as the agreement dimension in RE. This dimension is concerned with disparities and inconsistencies between different views of different stakeholders such as systems owners, customers, end-users, managers, requirements engineers, developers, and so forth. It deals with communicative and social activities among these people to support the resolution and integration of different viewpoints (Checkland & Scholes 1990; Goguen 1994; Coughlan & Macredie 2002). Therefore, problems and solutions "*exist*" in the eye of the beholder(s) and are grounded in their subjective views.

WORLD VIEWS

Why use metaphors like *problem solving* at all, if their fit with engineering activities is so poor? The explanation might be related to underlying world views dominating engineering. World views are the basis of all our models, understanding, and knowledge, and as such also the basis of our scientific approaches. Names for world views differ, and world view categorisations are overlapping: There is no objective way of describing all our world views, and, by the very nature of these views, there never will be. But with respect to academic life, acceptance of world views is split into two distinct groups. Gould (2000) calls one group *realist* and the other group *relativist*. The *realist* group is also linked to *positivist*, *objectivist*, *rationalist* and other related views, the *relativist* group is linked to *interpretivist*, *subjectivist* and other views. These views are all distinct from each other.

Their emphasis lies on different core beliefs and goals. However, for the purpose of this article, we will loosely group them into the *realist* and *relativist* camp.

By virtue of their positivist scientific education, and by virtue of rationalist engineering practices, engineers judge their experiences and their own contributions within a positivist/rationalist framework. In the case of the problem solving metaphor, one can identify both positivist and rationalist elements: The notion of *problem space* can be identified with the positivist view of pre-defined reality, and the understanding of the solving act with the rationalist view of improving or controlling conditions by applying the power of rational intervention.

Of course engineers are not entirely restricted by their world views. Thinking style and world view are related, but different. A relativist world view would appear to support divergent thinking more than a realist world view, by virtue of its inherent acceptance of subjective understanding and context-relative re-interpretation. But in practice adoption of relativist views indirectly motivates developing convergent thinking: transcending individual differences and arriving at shared understanding will be even more important for relativists, since he would have to produce trans-subjective meaning rather than assuming its a priori existence. But eventually, due to fundamental differences in implicit assumptions, alternative world views would lead to completely different models of information systems creation and use. The interpretivist view favoured in the humanities would for instance explicitly stress the subjective knowledge of the human stakeholders involved. These alternative views do find their way into information systems research, but they occupy niches left over by the dominant view, and research results compatible with these alternative views do not lead to a re-interpretation of core realist engineering models. Therefore, notions like *problem solving* are upheld in spite of their deficiencies.

Is it, then, time for a *paradigm shift* (Kuhn 1962) towards relativist models? Unfortunately, replacing the present view with another will not improve our practices, or the power of our understanding. It will only replace one area of weakness with another. The realist views have been adopted in science/engineering for good reason: They are the views best adapted to understanding nature/engineering. Replacing them will not improve our situation.

Merging relativist views with realist elements is also not a real option: World views are self-sufficient, they offer a complete set of beliefs, a framework for understanding any situation. As mentioned above, world views can not be proven wrong. As the literature on RE demonstrates, models based in a particular view can always be extended to absorb inconsistent empirical results: extending the rationalist notion of search in a problem space to that of an ill-defined wicked problem space that transforms itself to accommodate subjective perception and relies on problem solver intuition is already a substantial extension, and there is no reason to believe this notion cannot be stretched further.

In fact, the expectation of functional creativity placed upon engineers leaves no other choice but upholding notions of problem definition, problem solving and objective functionality. So the predominant views are justified, necessary, and self-sufficient. Some alien elements (like stakeholder's subjective knowledge) might be imported, but only as add-ons to the existing models. Merging entirely heterogeneous world views scientifically is not feasible. But as much as these views exclude each other in an ontological sense, we are able to compare and synthesise them as individual observers - as we just did. Coming back to the initial analysis of creative information systems use, we can detect expectations grounded in different world views:

In IS behaviour criterion 1, the dominant world view is clearly relativist: Multiple views incorporating multiple forms of knowledge representation can be used without restriction,

implying that the relationship between knowledge and knowledge representation is flexible as well. On the other hand, the computer is expected to somehow enable the switch from one type of knowledge representation to another, implying an objective way of translating between these representations, firmly grounded in realist expectations. A similar relativist-realist contradiction is true for IS behaviour criterion 2: relativist freedom to mix and match tasks at will is contrasted with the realist expectation of keeping individual tasks and knowledge interaction consistent.

As a mirror image of the above, IS behaviour criterion 3 is primarily realist, but with relativist undercurrents: Exploring implications of existing knowledge representations and evaluating them implicitly assumes a realist link between knowledge representations and knowledge, but dynamically including heterogeneous knowledge sources or new constraints implies relativist freedom in linking subjects, knowledge, and goals.

CONCLUSION

We have emphasized in this paper that information systems and creativity involves both engineered products and human players. The differences between the two are so significant as to justify using two different sets of world views to describe them. However the attempt to reconcile these two world views leads to a paradox: innovative thinking is grounded in qualities best understood in relativist terms, but information systems as engineering products are best understood in realist terms. Processes can best be understood in terms of relativist transformation of both problem and solution, but results are best understood in realist terms like precision and accuracy.

In this paper we have begun to work out the pathway towards resolving the paradox: In order to be of benefit to both information systems engineering and information systems use, our models of creativity will have to be capable of switching from realist to relativist views and back again. Unfortunately, this involves a trade-off: information systems cannot be expected to deliver perfectly reliable realist services while, at the same time, leaving maximum relativist freedom. The task is to find a balance. Creativity of the developer does not necessarily benefit or inhibit potential creativity of the user. In some situations, flexibility or other information system properties can be expressed in realist terms. In other situations, abandoning realist expectations and leaving relativist choice for later stages is more appropriate. In conceptualising, designing and building a bridge, for instance, what we have called "*reliable realist services*" may be far more important than relativist considerations. In the case of the Sydney Opera House, however, experience has shown that a relative lack of exact detail in specifying the task has ultimately produced a building that is almost universally praised, despite the fact that the original requirements were not met. Whenever realist expectations are maximised, information systems designers will have to take control of the process, and information systems user creativity will be restricted. Where user creativity is maximised, realist expectations will have to be reduced.

In order to address the socio-technological complexity of information systems and their use, pluralist research approaches and practices are required. From the point of view of education, this means that information systems designer creativity can be improved by teaching requirements engineers how to leave the door open for relativist freedom in later stages of engineering, and information systems user creativity can be improved by teaching information systems designers how to offer relativist freedom to users. Finally, because of the trade-off mentioned above, information systems usability for creative users can also be increased by teaching them to make tough choices in relation to their own expectations and

their own role in the production process. The design of educational experiences that will achieve these goals (teaching and learning methods, assessment procedures, and so on) now need to be considered.

REFERENCES

- Armarego, J. (2004), "Learning requirements engineering within an engineering ethos", in **Proceedings of 9th Australian Workshop on Requirements Engineering AWRE'04**, Adelaide, Australia.
- Batra, D. & Davis, J. G. (1992), "Conceptual data modelling in database design: similarities and differences between expert and novice designers", **International Journal Man-Machine Studies**, vol. 37:pp. 83–101.
- Beynon, M., Ch'en, Y.-C., Hseu, H.-W., Maad, S., Rasmequan, S., Joe, C., Rungrattanaubol, J., Russ, S., Ward, A., & Wong, A. (2001), "The computer as instrument", in M. Beynon, C. L. Nehaniv, & K. Dautenhahn, eds., **Cognitive Technology: Instruments of Mind : 4th International Conference, CI 2001**.
- Biggs, B. J. & Moore, P. J. (1993), **The Process of Learning**, Prentice Hall, Englewood Cliffs (NJ).
- Bruner, J. S. (1962), "The conditions of creativity", in H. Gruber, G. Terrell, & M. Wertheimer, eds., **Contemporary approaches to cognition**, pp. 1–30, Atherton, NY, USA.
- Burghardt, M. D. (1995), **Introduction to the engineering profession**, Addison Wesley, NY, USA.
- Candy, L. & Edmonds, E. (1997), "supporting the creative user: a criteria-based approach to interaction design", **Design Studies**, vol. 18(2):pp. 185–194.
- Carroll, J. & Swatman, P. A. (1999), **Opportunism in the Requirements Engineering Process**, School of Information Systems, Deakin University.
- Checkland, P. & Scholes, J. (1990), **Soft Systems Methodology in Action**, John Willey & Sons Ltd, Chichester, UK.
- Committee on Information Technology and Creativity (2004), **Beyond Productivity**, National Academy Press.
- Coughlan, J. & Macredie, R. D. (2002), "Effective communication in requirements elicitation: A comparison of methodologies", **Journal of Requirements Engineering**, vol. 7(2):pp. 47–60.
- Coyne, R. (2005), "Wicked problems revisited", **Design Studies**, vol. 26:pp. 5–17.
- Cropley, A. J. (2006), "In praise of convergent thinking", **Creativity Research Journal**, vol. 18.
- Cropley, D. H. & Cropley, A. J. (2000), "Fostering creativity in engineering undergraduates", **High Ability Studies**, vol. 11:pp. 207–219.
- Cropley, D. H. & Cropley, A. J. (2005), "Engineering creativity: A systems concept of functional creativity", in J. C. Kaufman & J. Baer, eds., **Faces of the muse: How people think, work and act creatively in diverse domains**, pp. 169–185, Lawrence Erlbaum, Hillsdale, NJ, USA.
- Csikszentmihalyi, M. (1997), **Creativity: Flow and the Psychology of discovery and invention**, HarperPerennial, NY, USA.
- Cybulski, J., Nguyen, L., Thanasankit, T., & Lichtenstein, S. (2003), "Understanding problem solving in requirements engineering: Debating creativity with its practitioners", in **PACIS 2003 Proceedings of the Seventh Pacific Asia Conference on Information Systems**,

- Adelaide, Australia.
- Dallman, S., Nguyen, L., Lamp, J., & Cybulski, J. (2005), "Contextual factors which influence creativity in requirements engineering", in **Proceedings of 13th European Conference on Information Systems ECIS 2005**, Regensburg, Germany.
- Edmonds, E. & Candy, L. (2002), "Creativity, art practice, and knowledge", **Communications of the ACM**, vol. 45(10):pp. 91–95.
- Edwards, S. M. (2001), "The technology paradox: Efficiency versus creativity", **Creativity Research Journal**, vol. 13(2):pp. 221–228.
- Flavell, J. (1979), "Metacognition and cognitive monitoring: a new area of cognitive-developmental inquiry", **American Psychologist**, vol. 34(10):pp. 906–911.
- Gause, D. C. & Weinberg, G. M. (1989), **Exploring Requirements: Quality Before Design**, Dorset House Publishing, New York, USA.
- Goguen, J. A. (1994), **Requirements Engineering: Social and Technical Issues**, chap. Requirements Engineering as the Reconciliation of Social and Technical Issues, pp. 165–199, Academic Press, London, UK.
- Gould, S. J. (2000), "Deconstructing the 'science wars' by reconstructing an old mold", **Science Magazine**, vol. 287(5451):p. 253.
- Guilford, J. P. (1950), "Creativity", **American Psychologist**, vol. 5:pp. 444–454.
- Guindon, R. (1990), "Designing the design process: Exploiting opportunistic thoughts", **Human-Computer Interaction**, vol. 5:pp. 305–344.
- Hadamard, J. (1954), **The Psychology of Invention in the Mathematical Field**, Dover Publications, New York, USA.
- Hewett, T. T. (2005), "informing the design of computer-based environments to support creativity", **International Journal of Human-Computer Studies**, in press.
- Higgins, J. M. (1994), **101 Creative Problem Solving Techniques: The Handbook of New Ideas for Business**, New Management Publishing Company, Winter Park, FL, USA.
- Horenstein, M. N. (2002), **Design concepts for engineers**, Prentice Hall, Upper Saddle River, NJ, USA.
- Khushalani, A. J. (1997), **Modelling and Supporting Opportunistic Design Problem Solving**, Ph.D. thesis, School of Computer Science and Software Engineering, Swinburne University of Technology, Melbourne, AU.
- Kuhn, T. S. (1962), **The Structure of Scientific Revolutions**, The Chicago University Press.
- Lawson, B. (1999), "'fake' and 'real' creativity using computer aided design: Some lessons from herman hertzberger", in **Proceedings of Creativity & Cognition 1999**, pp. 174–179.
- Loucopoulos, P. & Karakostas, V. (1995), **System Requirements Engineering**, McGraw-Hill Book Company, New York, USA.
- Maiden, N. & Gizikis, A. (2001), "Where do requirements come from?", **IEEE Software**, vol. 18(5):pp. 10–12.
- Maiden, N., Manning, S., Robertson, S., & Greenwood, J. (2004), "Integrating creativity workshops into structured requirements processes", in **Proceedings of the 2004 conference on Designing interactive systems**, ACM Press.
- Mich, L., Anesi, C., & Berry, D. M. (2004), "Requirements engineering and creativity: An innovative approach based on a model of the pragmatics of communication", in **Proceedings of Requirements Engineering: Foundation of Software Quality REFSQ'04**, Riga, Latvia.
- Morning Herald (1904), "Adelaide university examinations".
- Nguyen, L., Carroll, J., & Swatman, P. A. (2000), "Supporting and monitoring the creativity of is personnel during the requirements engineering process", in **Proceedings of 33rd**

Hawai'i International Conference on System Sciences HICSS-33.

- Nguyen, L. & Swatman, P. A. (2000), "Essential and incidental complexity in requirements models", in **Proceedings of 4th IEEE International Conference on Requirements Engineering ICRE 2000**.
- Nguyen, L. & Swatman, P. A. (2003), "Managing the requirements engineering process", **Requirements Engineering**, vol. 8(1):pp. 55–68.
- Nguyen, L. & Swatman, P. A. (2006), **Rationale Management in Software Engineering**, chap. Promoting and Supporting Requirements Engineering Creativity, Springer.
- Ohlsson, S. (1984), "Restructuring revisited: Summary and critique of the gestalt theory of problem solving", **Scandinavian Journal of Psychology**, vol. 25:pp. 65–78.
- Pennel, L. & Maiden, N. A. M. (2003), "Creating requirements: Techniques and experiences in the policing domain", in **REFSQ'2003 Workshop Proceedings**, Velden, Austria.
- Pohl, K. (1994), "Three dimensions of requirements engineering: A framework and its application", **Information Systems**, vol. 19(3):pp. 243–258.
- Resnick, L. B. (1987), **Education and learning to think**, National Academy Press, Washington, DC, USA.
- Robertson, J. (2002), "Eureka! why analysts should invent requirements", **IEEE Software**, vol. 19(4):pp. 20–22.
- Shanks, G. & Simsion, G. (1992), "Automated support for creative database design", in B. Srinivasan & J. Zeleznikov, eds., **Research and Practical Issues in Databases**, pp. 293–305, World Scientific Publishing.
- Shneiderman, B. (2002), "Creativity support tools", **Communications of the ACM**, vol. 45(10):pp. 116–120.
- Sternberg, R. J. & Lubart, T. I. (1992), **Creative giftedness in children**, Ablex, Norwood, NJ, USA.
- Visser, W. (1990), "More or less following a plan during design: Opportunistic deviations in specifications", **International Journal of Man-Machine Studies**, vol. 33:pp. 247–278.
- Voland, G. (2004), **Engineering by Design**, Pearson Education, Upper Saddle River, N J, USA.
- Wallas, G. (1926), **The Art of Thought**, Jonathan Cape.