Industrial Engineering The University of Iowa

Working Paper No. ISL_01_10_2005

Innovation Science: A Primer

Andrew Kusiak, Professor
Intelligent Systems Laboratory
Department of Mechanical and Industrial Engineering
2139 Seamans Center
The University of Iowa
Iowa City, Iowa 52242 - 1527

Tel: 319-335-5934 Fax: 319-335-5669
Email: andrew-kusiak@uiowa.edu
http://www.icaen.uiowa.edu/~ankusiak

Iowa City, IA October 2005

Innovation Science: A Primer

Andrew Kusiak, Professor
Intelligent Systems Laboratory
Department of Mechanical and Industrial Engineering
2139 Seamans Center
The University of Iowa
Iowa City, Iowa 52242 - 1527

Tel: 319-335-5934 Fax: 319-335-5669
Email: andrew-kusiak@uiowa.edu
http://www.icaen.uiowa.edu/~ankusiak

Abstract

The term innovation resonates broadly in cyberspace, books, and journals. A careful analysis of the vast open-source information indicates the engineering literature on the underlying science of innovation is limited. Innovations in any domain can be enhanced by principles and insights from different disciplines. However, the process of identifying the linkages between the diverse disciplines and the target domain is not well understood. The innovation process and conditions triggering innovation set the stage for economic progress. The paper contributes to better understanding of the process of innovation by introducing basic innovation models. The ideas outlined in the paper provide a roadmap for areas of future study as innovation science can provide a pathway for industries to be able to successfully compete in the global market.

Keywords: *Innovation science, innovation process, data mining, knowledge discovery, evolutionary computation.*

1. Introduction

A product, process, service, or a business can be described with various metrics, e.g., cost, quality, reliability. The emerging metric of particular interest is innovation. Piand (2003) described innovation as the activity of people and organizations to change themselves and the environment. The latter implies breaking a routine way of thinking and using new approaches. The scope of innovation varies from product and process to organization or even a society. The nature of innovation is user dependent, e.g., a product innovation for a designer can be a process innovation for a manufacturer.

The 21th century customers are better informed than ever before. The interaction time between a customer and a product has reduced. Companies are forced to analyze customer needs and behaviors impacting the product success in the marketplace.

Innovation in a manufacturing environment is often expressed in the literature as a function of uncertainty between a product and a process as illustrated in Fig. 1.

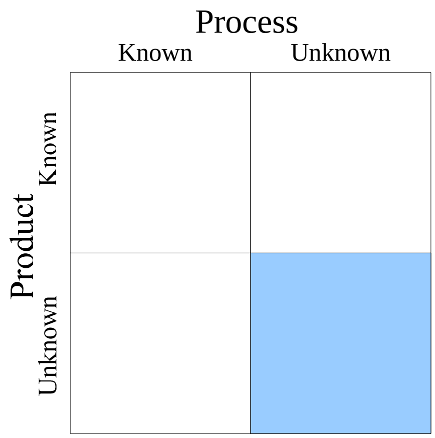

Figure 1. Innovation quadrant.

The lower right quadrant (shaded) in Fig. 1 involves unknown (high uncertainty) process and unknown product and therefore is considered as the quadrant with the highest innovation potential. However, innovation may take place in any of the remaining three quadrants. Thus each quadrant represents innovation of a different nature and scope. For example, innovation in the upper left quadrant may involve a subassembly or even a component.

The study of innovation – the development of new knowledge and artifacts – is of interest to engineering, business, social and behavioral sciences, and spans sociology, history, philosophy, economics, psychology, and political science (Troyer 2005). Innovations transform economies (e.g., California's agricultural economy transformed into the knowledge-based Silicon Valley economy). Innovations alter global relations (e.g., the impact of nuclear technologies on international treaties), and produce new structures of social control (e.g., the creation of international regulatory agencies to oversee pharmaceutical industries). Innovations change the day-to-day lives of individuals (e.g., the development and introduction of new biopharmaceutical discoveries that affect quality of life).

Innovations in any domain can be enhanced by principles and insights from other disciplines. However, the process of identifying the linkages between different domains and the need for innovation science is apparent. Innovations of products and processes are of particular interest to manufacturing and service applications.

1.1 Why Innovation Science?

There is a growing consensus among industry and academia that innovation should be studied. There are several new initiatives that address innovation. Two of the more prominent ones include: (1) the report *Innovate America*, published by the Council on Competitiveness (NIIR 2004), and (2) the Center for the Study of Innovation and Productivity launched by the San Francisco Federal Research Board (`http://www.EconData.net`) to study innovation, technology and productivity and their contributions to economic activity. Some of the facts that warrant accelerated development of innovation science include:

- Innovation is the engine of the global economy, accounting for some 50% of the economic growth (NIIR 2004).

- Innovation will mark the first economic revolution of the 21st century (Shah 2004).
- Innovation involves almost all aspects of life, yet the innovation process is not well understood.
- Innovation applies to the creation of methods used in industry, including the design of consumer goods, defense products, medical devices, medications, and services.
- The increasing complexity of technologies, their interdependencies, and the rapidly expanding volume of data call for a paradigm shift to be led by innovation.
- Educational revolution, in particular in engineering, is needed to create innovative workforce.

Innovation has been studied by psychologists and group process researchers at multiple levels, including the organizational level. Researchers have investigated how alternative leadership styles, varying degrees of worker autonomy, and organizational cultures (i.e., systems of values, norms, and beliefs) affect innovation in R&D teams (e.g., Cohen *et al.* 1982; Troyer 1995, 2004).

There are several areas where the study of innovation could initiate and potentially formalize the science of innovation. This includes the study of existing literature and patents, and innovators and creators (e.g., musicians, painters). Based on these studies one can conceptualize and model the innovation processes and its generalizations across engineering, arts, science, and social domains.

1.2 Basic Typology of Innovation

The industry has used three basic approaches to innovate: structured, creative, and dynamic, producing either a sustaining or disruptive product referred to as innovative (Allen 2003). Structured innovation spawned during the industrial era, was engineered to be highly efficient and replicable by innovating within set guidelines. It has been primarily used in large corporations, and it emphasizes internal leadership, strategic planning, effective execution of ideas, shareholder pressure, and financial resources more than other approaches, while placing less emphasis on a creative environment (Report 2003). Creative innovation thrives more often in small organizations where focusing on "the big picture" can be accomplished more easily as these companies tend to consider the inspirational aspects of innovation versus the process (Allen 2003; Shah 2004). The greatest advantage to the creative approach is the process itself (Report 2003). Dynamic innovation is a blend of both the structure and creative innovation approaches. Businesses of all sizes from small to large have used the dynamic approach to produce successful innovation. Dynamic innovation has taken on the aspects of structured innovation that embody strategic thinking and planning, along with the need for execution of projects. Dynamic innovation incorporates cross-functional collaboration and makes the senior executive in charge of the innovation in the company. Even though 36% of participating companies have adopted this method, most of them would rank it as high risk (Report 2003).

Sustaining innovations are built off previous innovations (Allen 2003), e.g., the palm PDA. The PDA been an innovative and successful device, however, its predecessor the Apple Newton has failed. Sustaining innovations tend to be more successful then the disruptive innovations. The reason for this is that sustaining innovations are built based on a product or a process that is known to the market. The sustaining innovation is easier to develop and market, as it follows the incumbent.

Disruptive innovations are referred to as paradigm-shifters. They make current standards obsolete and anticipate future needs (Allen 2003). In the past, the heuristic rule was that a disruptive innovation occurred once every few decades, e.g., electricity, steam engines, assembly lines. Nowadays, innovations are brought to market more frequently, e.g., yearly. The example mentioned of a disruptive technology, Apple Newton, was large, bulky and not user friendly. Disruptive innovation is often not profitable, since it is expensive to develop and market. Some corporations do not invest in disruptive innovations due to the increased risk of losses.

2. Product Requirements and Innovation

The past two decades have seen the customer perspective reflected mostly in the product function and form. In the 1980th the interest has begun to shift from the requirements defined by experts (often design engineers) to the customer defined requirements. This customer focus has been driven by the necessity to increase customer satisfaction. The commonly used attributes used to measure customers' satisfaction often involved quality, reliability, and cost. The broadly accepted industrial initiatives such as concurrent engineering, integrated product and process design, and kaizen programs, have taken as serious look at the customer oriented attributes in the design of new products.

Product innovation I can be expressed as a function of requirements $\boldsymbol{x}$, $I = f(\boldsymbol{x})$. Understanding the requirements is key to the design of innovative products.

The more sophisticated and informed customer has imposed higher expectations on the product. A customer of today not only wants to get a product he/she perceives (product personalization), but is also impacted by additional attributes such as surprise (e.g., unexpected product function), pleasure (e.g., driving a car), fantasy, and so on. The list of these new requirements has not been completely defined; rather it evolves in time (see Fig. 2).

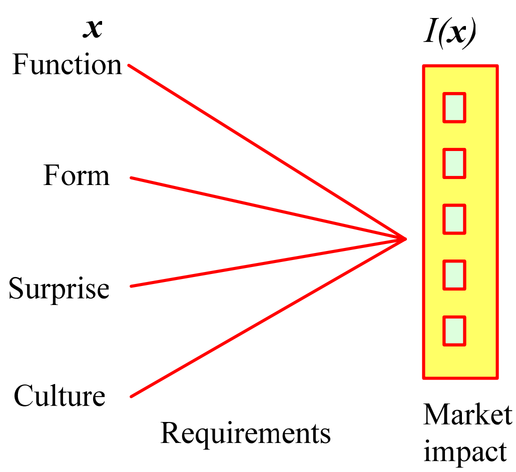

Figure 2. Expanded list of requirements.

One will likely see new product attributes (introduced by new requirements) emerging in time. They will be reflected in the product designs and used in marketing to attract new customers. It will take multidisciplinary research to develop better understanding these attributes and matching them with the product development programs.

An innovative design may emerge from the previous generations of the same product by considering new requirements. The innovation problem can be then reduced to the requirements formulation problem. An attempt should be made to capture the innovation-prone requirements as early as possible, ideally at the requirements formulation phase (Design phase 0 in Fig. 1) of the design process. One should also realize that additional requirements can be generated later in the design process (see Fig. 3). In fact any alteration of the existing and new requirements may take place along the product development life-cycle.

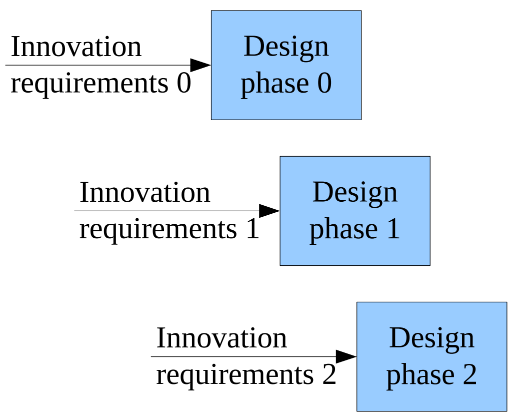

Figure 3. Context and time dependent innovation engine.

An open question that deserves separate investigation is how much of innovation happens outside of the requirements fostering innovation. The answer to this question is not easy as historical data and examples that could support or reject this hypothesis may not be easily available. One could argue, however, that even if the innovative aspect of the design has been conceived without a previously formulated requirement such a requirement could be generated when a serious attempt to create it would have been made.

Though many of the innovation issues included in this paper are discussed in the context of product design, they equally apply to the design and creation of processes and services. Using the proposed approach to generate hybrid solutions, e.g., a product, a process, and a service supporting the product could be the greatest asset.

2.1 Definition of refined requirements

There are numerous ways of eliciting detailed requirements:

- Traditional customer surveys and user-based input
- Data analysis, in particular data mining (e.g., Kantardzic 2003)
- Evolutionary computation tools, in particular genetic programming discussed later in this paper.

Any approach producing requirements leading to product success is commendable. The focus of this research is to explore formal approaches to the generation of requirements, especially such requirements that are likely to produce innovative designs. Examples of two approaches that naturally fit here are data mining and evolutionary computation. They could be used independently or work in tandem.

Data-mining algorithms discover patterns in the data that may transform into requirements of interest. Since the width of data analyzed by the data-mining algorithms is practically unlimited, the patterns are likely to be unanticipated and interesting. The value delivered by these patterns is strictly related to the quality of data and textual databases used for mining. Besides the comprehensiveness of data processing, data mining brings yet another advantage – it may be used to support the needs of an individual customer.

3. Innovation Science Research

Scholars of technology have indicated that innovation lies at the intersection of science and technology (e.g., Pinch and Bijker 1990). Within this perspective, "technology" is synonymous with "applied science" (i.e., the production of goods and services based on scientific research). One view proposes that innovation becomes possible through advances in basic science (e.g., the development of new ideas and theories) and is realized in concrete products within the context of applied science. Another view suggests that the development of innovative products through applied science generates new resources on which basic science draws to advance new ideas and theories (Troyer 2005). Barnes (1982) has proposed that science and technology are enjoined in a symbiotic relationship, drawing from and contributing to one another's cultures. The symbiosis, however, may not always involve facilitative relations. Interactions between basic and applied scientists are often characterized by conflict arising from different research methods and strategies, status tensions, and differences in occupational cultures (e.g., Haribabu 2000).

As a new science, innovation is likely to borrow concepts from the existing sciences, e.g., data mining, evolutionary computation, cognitive sciences. Creativity and innovation are often considered as inseparable (Sternberg 2005). In fact, the breadth of the science base of innovation is likely to be larger than any of the known sciences.

The following five models of interest to innovation science are discussed next:

1. Hypothesis-based model
2. Optimization-based model
3. Evolutionary computation model
4. Pattern discovery model
5. Process model

3.1 Hypothesis-Based Model

The innovation science should look at the role of hypothesis driven vs hypothesis discovery research. A framework for maintaining the proper balance between the two should be established. Hypotheses fostering innovation may have different ownership. In the product design context they can be generated by the customers, marketing departments, or the designers themselves. The growing volume of data collected along the product life-cycle and the information about the customers warrants a hypothesis-based discovery approach to be supported by data mining. The difference between the two approaches is highlighted in Fig. 4.

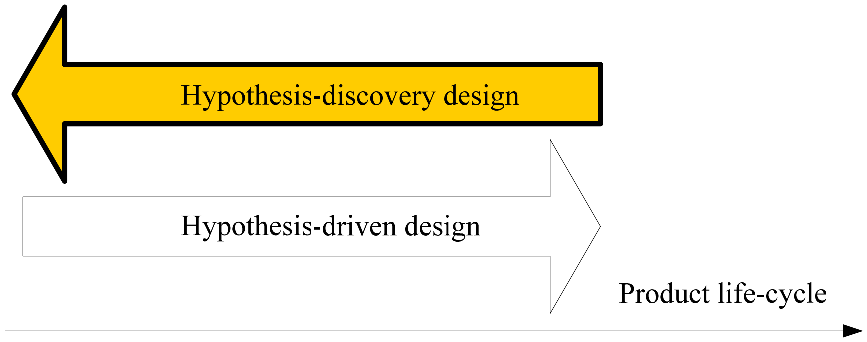

Figure 4. Information flow.

The emerging hypothesis-discovery approach changes the direction of information flow along the product-life cycle.

3.2 Optimization-Based Model

An optimization model of innovation involves objective function and constraints. For example, consider the innovation function in Fig. 5. Maximizing the innovation function $I = f(x)$ subject to a constraint $1 \leq x \leq 3$ would produce a local maximum, however, relaxing this constraint to $1 \leq x \leq 6$ could result in a global maximum. Modifying the same constraint to $4.5 \leq x \leq 5.5$ would be equivalent to a targeted innovation, where a reasonable effort (represented by the computation needed to determine the maximum of the function $I = f(x)$ (in Fig. 5) would maximize the innovation impact.

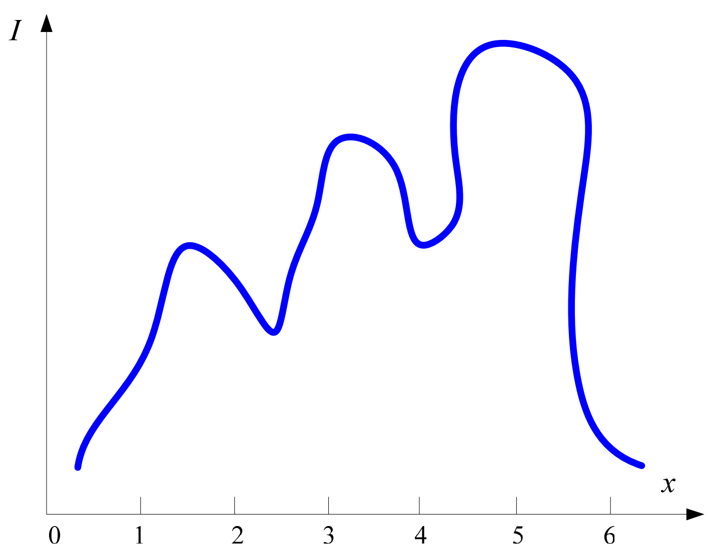

Figure 5. Innovation function $I = f(x)$.

The optimization model of innovation is generalized by the evolutionary computation framework, in particular genetic programming discussed next.

3.3 Evolutionary Computation Model

Evolutionary computation deals with models based on natural evolution. A number of evolutionary computational algorithms have been developed, including genetic programming (e.g., Koza 1992, 1994 and Benzhaf *et al.* 1998), evolutionary algorithms (e.g., Coello 1999), evolutionary strategies (e.g., Eiben and Smith 2003), and artificial life (e.g., Engelbrecht 2003).

The applicability of evolutionary computation to innovation science is illustrated with genetic programming.

What is genetic programming?

Genetic programming is an algorithm that can be used in a variety of ways to process data (Koa 1992). The proposed use of genetic programming in requirements-based innovation is to generate unexpected combinations of requirements, product functions, or product architectures. Besides functioning on its own, the genetic programming algorithm could be used in conjunction with data mining.

Genetic programming uses biologically inspired operations such as reproduction, crossover, and mutation, which are similar to those used in genetic algorithms. In addition it involves architecture-altering operations, more general solution representation schemes, and more rich operators than those of genetic algorithms.

The main steps of a genetic programming algorithm include (Koza 1992, 1994]:

Creation of Initial Population of Solutions

Functions and terminals are used to generate a random population of initial solutions. The set of functions may include arithmetic functions and conditional operators. The set of terminals include external inputs (such as the features) and random constants (such as 5.10 and 44.35). The randomly created initial solutions are typically of different sizes and shapes.

Main Loop of Genetic Programming Algorithm

The main loop of genetic programming includes fitness evaluation, selection, and genetic operations. The fitness of each individual solution in the population is evaluated. Solutions are then probabilistically selected from the population based on their fitness to participate in the various genetic operations, with reselection allowed. While a solution that is fit may have a better chance of being selected, unfit individuals compete. After numerous generations, an acceptable solution emerges.

Mutation Operation

The mutation operation selects probabilistically a single parental solution from the population based on the fitness value. A mutation point is randomly chosen, the partial solution rooted at that point is deleted, and a new partial solution is grown according to the same random growth process that was used to generate the initial population.

Crossover Operation

In the crossover, two parental solutions are probabilistically selected from the population based on the fitness value. The two parents participating in crossover are usually of different sizes and shapes. A crossover point is randomly chosen at the first parent and a crossover point is randomly chosen at the second parent. Then the partial solution at the crossover point of the first parent is deleted and replaced by the partial solution from the second parent. The crossover operator is dominant in genetic programming.

Reproduction Operation

The reproduction operation copies a single individual solution, probabilistically selected based on fitness, into the next generation of the population.

Structure-Altering Operations

Rather than using a user-specified fixed structure for all solutions in the population, genetic programming allows for structure-altering operations to automatically determine solution structure that correspond to the natural gene transformations. These structure-altering operations produce population containing architecturally diverse solutions.

While most steps of the genetic algorithm appear to be feasible for implementation in innovation-driven product design, construction of the fitness function and its evaluation methods are not easy. For example, consider the design of modular products with a set of predefined components. Writing a computer program to evaluate the different part configurations appears to be difficult, especially in mechanical design. Representing an internal solution produced by the genetic algorithm with geometry would certainly ease this evaluation. For example, consider the visual evaluation of the fitness function illustrated in Fig. 6, where the genetic programming solution (GP) is expressed with geometry (a phenotypic expression). The quality of the geometry (design) is evaluated by a human user and the feedback is provided to the genetic programming algorithm.

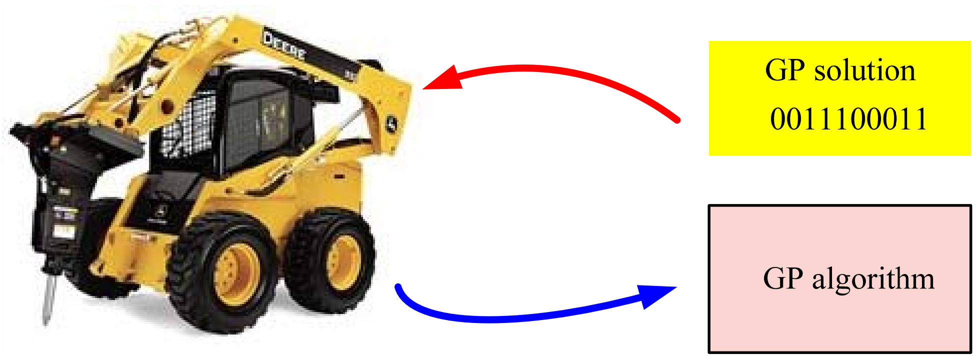

Figure 6. Phenotypic evaluation of the fitness function (The product design is courtesy of Deere & Company, Moline, IL).

The geometric evaluation of the fitness function shown in Fig. 6 is one of many possible ways of providing feedback to the GP algorithm.

3.4 Pattern Discovery Model

The role of patterns in innovation offers a great potential especially as large volumes of data become available. The data with potential impact on design is collected prior, during, and after the product has been designed. In essence, the design of a product is embedded in the data space containing knowledge pertaining to different aspects of the design, including innovation.

The patterns discovered in data may change the design paradigm form an open-loop system (see Fig. 7) to the closed-loop system shown in Fig. 8.

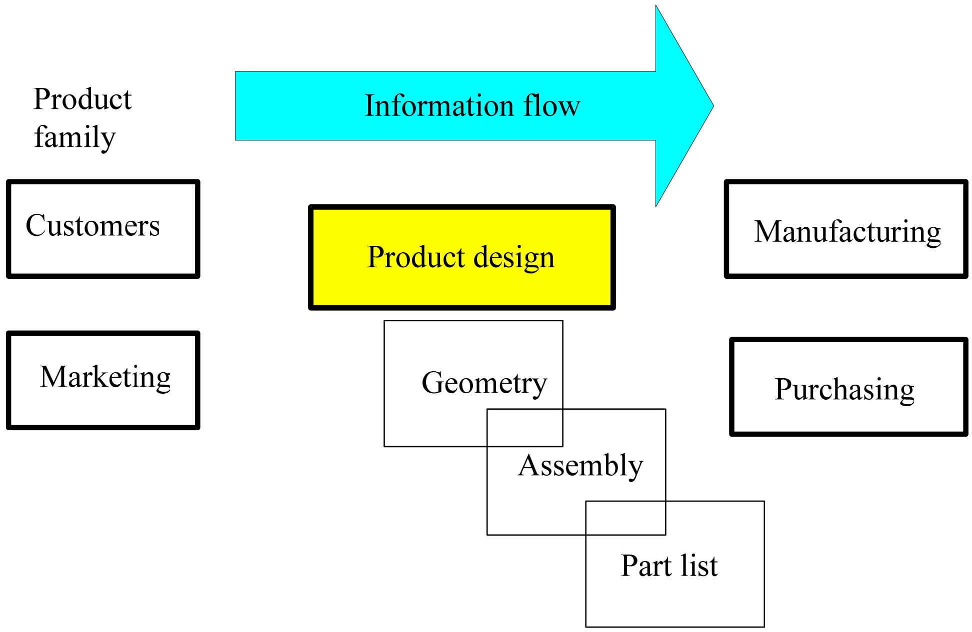

Figure 7. An open-loop design system.

Tradition design is essentially an open-loop design with the information flowing in one direction.

The patterns discovered from the data provide a valuable feedback to the design. Thus an open-loop design system becomes a closed-loop design system (Fig. 8). Though the scope of patterns in design may be large, innovation may be one of the greatest beneficiaries of the pattern discovery with data-mining algorithms.

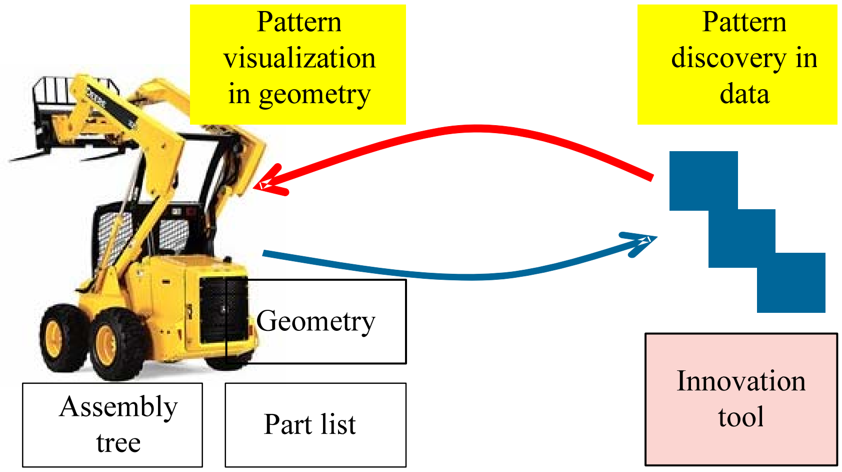

Figure 8. The innovation-in-the-loop design system (The product design is courtesy of Deere & Company, Moline, IL).

The data-mining extracted patterns can be descriptive (e.g., formed with clustering algorithms) and predictive (e.g., extracted with decision-rule algorithms). Examples of the two types of patterns are discussed next. The forming of descriptive patterns is illustrated with the dependency structure matrix and clustering, while the predictive data mining is illustrated with decision rules.

Dependency Structure Matrix

In traditional design of products and product families, only limited interactions have been considered, mainly spatial interaction, energy, information, and material (Browning 2001). Physical proximity, alignment, and orientation were the underling reasons for defining these interactions. A frequent use of this interaction information would be modularity defined by the concept of the dependency structure matrix (DSM) (Steward 1981, Kusiak and Wang 1993, Ulrich and Eppinger 2000). The traingularization algorithm (the matrix reorganization algorithm) discussed in Kusiak *et al.* (1994) derives the interaction patters by transforming the dependency-structure matrix from an unstructured form to the form that most resembles the lower triangular matrix.

Clustering

Innovation calls for expanded definition of interactions, and determining a variety of patters. All patterns can be important, irrespectively of the type of interactions among them. Components that interact directly are candidates for modules (called here physical modules), while parts with no physical proximity and interactions form logical (virtual) modules. For example, if the same type and size tires and the steering wheels (a logical module) would be used across 95% of the in the designed vehicles, they would likely be assembled on the vehicle in the factory. However, tires of 20 different types and 25 stirring wheels would be likely mounted at the car dealership. The information present in the patterns can be used in different ways. The close proximity information is likely to be utilized at the product design stage (physical module design). However, the logical modules can be implemented in a number of ways, e.g., as late product differentiators at the product assembly stage, or a sales outlet. Some of the component interactions discovered with data mining that may appear to be incidental could in fact be a source of innovation. For example, the vehicle could be steered with a mechanism different that the steering wheel.

Clustering algorithms form groups of objects that share common properties. The early cluster analysis algorithms are the *k*-means algorithm, ISODATA, and the quick partition algorithm (Anderberg 1973). Cluster analysis algorithms falls into the category of unsupervised classification tools. For review of most recent cluster analysis algorithms see Han and Kamber (2001) and Kusiak (2000).

The computational intelligence community has studied conceptual clustering (Michalski 1983) as well as other methodologies with a statistical flavor. The basic idea behind conceptual clustering is that instead of considering the similarity between objects, conceptual cohesiveness among the objects is considered as a criterion for classification. Conceptual clustering techniques are context based and arrange objects hierarchically (Michalski 1983).

Autoclass is a known Bayesian classifier proposed by Cheeseman *et al.* (1988). Their strategy involved making simplifying assumptions about the classification model. Rather than searching the entire hypothesis space and considering all states, they focused on a limited number of possible states thereby reducing the number of possibilities to be analyzed. In the case of real value attributes, the assumption is that data is distributed according to the normal probability distribution. A multinomial distribution is assumed for the discrete attributes. Autoclass uses the expectation maximization (EM) algorithm (Dempster 1977), to estimate the class parameters that maximize the posterior probability of the parameters for a given number of classes. The Autoclass algorithm can be downloaded from the NASA (2005) website.

Decision rules

Decision-rule and decision-tree algorithms belong to a large class of supervised learning algorithms generating explicit knowledge (patterns). The two classes of algorithms have been implemented in numerous ways, for example:

- Decision-tree algorithms (e.g., ID3 [Quinlan 1986], CN2 [Clark and Boswell 1989], C4.5 [Quinlan 1993], T2 [Auer *et al.* 1995], Lazy decision trees [Friedman *et al.* 1996], OODG [Kohavi 1995], OC1 [Aha 1992], AC, BayTree, CAL5, CART, ID5R, IDL, TDIDT, and PROSM [all discussed in Michie *et al.* 1994]).
- Decision-rule algorithms (e.g., AQ15 [Michalski *et al.* 1986], LERS [Grzymala-Busse 1997, and numerous other algorithms based on the rough set theory [Pawlak 1991]).

Structured rules

The decision rules extracted in data mining may be used in "as-is" form or be structured. Rule structuring (Kusiak 2000a), is to enhance interpretability of the knowledge generated with machine learning algorithms. The need for knowledge structuring is supported by the notion of cognitive maps and mental models discussed in Caroll and Olson (1987) and Wickens *et al.* (1998). By structuring decision rules a human dimension will be incorporated into the knowledge extracted from data. The idea of structured knowledge is introduced by two examples of simplified decision tables presented in Fig. 9. Each of the four decisions A – D in Fig. 9(a) is made based on the same number of features. A learning algorithm has derived each of the four decision rules based on 1,000 examples. There is no exception to these rules, which creates an ideal decision-making setting that could be easily automated. The decision-maker matches the features of a new decision case with the features in the decision table and assigns the new case a decision equal to one of the four decision rules represented in the table. For example, a new case with the feature values F1 = yes, F2 = 1, F3 = 1.9 would be assigned decision B by Rule 2 of Fig. 9(a). Note that in this table the decisions are differentiated based on the feature values, rather than the features themselves.

In the decision table in Fig. 9(b) the decisions A – D are differentiated on features. Each of the four decisions is made based on the values of three to four different features. The feature sets associated with each of the four rules and decisions are mutually exclusive.

(a)

	F1	F2	F3	Decision	Support
Rule 1	yes	1	[8.1-9.9]	A	1000 examples
Rule 2	yes	1	[1.7-2.1]	B	1000 examples
Rule 3	no	2	[2.2-4.9]	C	1000 examples
Rule 4	yes	2	[5.0-8.0]	D	1000 examples

(b)

	F1	F7	F10	F4	F6	F9	F3	F5	F8	F12	F2	F6	F11	Decision	Support
Rule 1	ax	2	[7.7-9.1]											A	1000 examples
Rule 2				[1.3-1.7]	bb	4								B	1000 examples
Rule 3							di	7	yes	no				C	1000 examples
Rule 4											no	4	[52-56]	D	1000 examples

Figure 9. Examples of simple decision tables: a) single feature table, b) multi-feature table.

Other cases of decision differentiation are possible and they will be studied in this research together with various structures of decision matrices.

Analyses of many engineering data sets indicate that in many cases decision tables have distinct structures. Exploring different structures of tables is helpful in decision making due to:

- Decision process becoming transparent to the user and computing environment
- features get exposed which is helpful in planning data acquisition.

The decision table in Fig. 10 illustrates the case where decisions are differentiated based on features and their values.

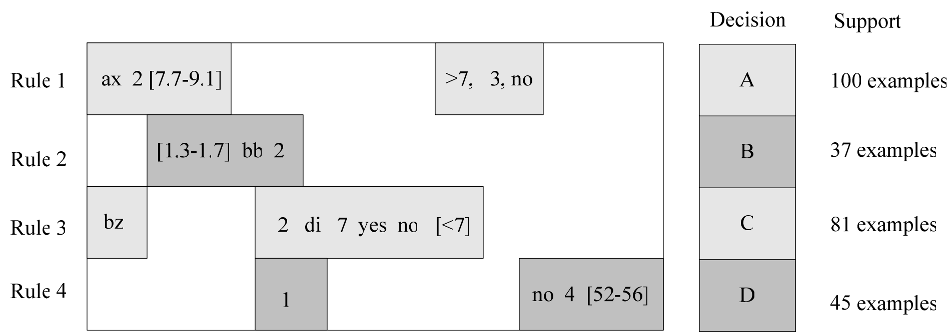

Figure 10. Example of a decision table with differentiation based on features and their values.

The decision table structure and the decision differentiation methods are determined by factors such as:

- Type of the learning algorithm
- Rule selection criteria
- Constraints and objective functions imposed on a decision table structure

The structured decision tables offer potential for multiple applications. They can serve as a backbone of a visualization environment (e.g., virtual reality) and increase transparency of the decision making process.

3.5 Process Model

Numerous methodologies have been developed for modeling processes. Although they vary in scope, representation, and theoretical foundations, each methodology provides insights from a particular perspective. Some of the existing process-modeling methodologies of interest to modeling innovation are listed next.

- *UML*: Unified Modeling Language is a visual and graphical modeling language to analyze and design object-oriented systems. Besides software development, UML can be used for process modeling. UML includes use case, sequence, collaboration, class, object, state, activity, component, and deployment diagrams (UML 2005).
- *CIM-OSA*: Computer Integrated Manufacturing - Open Systems Architecture. Four enterprise views are provided: function, information, resource, and organization (Beekman 1989).
- *GRAI Method*: This method is built around a conceptual reference model that is based on the theory of complex systems, hierarchical systems, organization systems, and the discrete activity theory (Doumeingts *et al.* 1987).
- *IDEF Methods*: A family of tools, including IDEF0 for functional modeling and IDEF3 for process modeling initiated by Air Force Program for Integrated Computer-Aided Manufacturing (Mayer *et al.* 1992).
- *IEM*: A public domain methodology designed around the object-oriented paradigm.
- *SSADM*: A method of systems analysis with the focus on the information perspective (Ashworth 1988).

A product development model (intertwined with innovation activities) involves activities that are not known in advance and are not well predicted. The uncertainty associated with the innovation activates calls for innovation process management (Tatikonda and Rosenthal 2000). Data-mining algorithms may be used to determine the underlying patterns of success. Though these patterns are likely to be temporal, any use of structures is helpful in the execution of the innovation process (Tidd *et al*. 2001).

To date numerous innovation models have been generated (Tidd *et al.* 2001). The early models viewed innovation as linear process with focus on either a technology-push or a demand-pull innovation process (Schwery and Raurich 2004). The prevailing view in the literature points to innovation models with complex interactions and cycles. The scope of innovation models has been widened to include suppliers, and business alliances, all serving customers demanding personalized products.

4. Innovation Enhancing Tools

Numerous tools have been developed in support of innovative design of products, including TRIZ (TRIZ Journal 2005), the creative problem solving (CPS) process (Daupert 2005), and the innovation technology (IvT) approach.

TRIZ was developed to foster innovation by analyzing the patterns of problems and solutions, rather than relying on the spontaneous creativity of individuals or groups (Domb 2003). This is done by focusing on a problem in its basic form while simultaneously understanding that the problem considered is rarely the one to be solved. TRIZ handles three basic problems: the technical conflict and physical contradiction problem in which a solution creates another problem; the inventive problem where before a problem is solved, the solution of the conflict must be resolved; and the creation of the ideal machine/process in which something simplistic is constructed from a concept (Siem 1996).

The CPS (Daupert 2005) is a problem solver for a generation of innovative solutions. During the solution generation process, combining convergent and divergent thinking is used to produce numerous potential solutions, while the user imagination is used freely to aid in the creation of innovative and working solutions.

Another approach used by engineers is the innovation technology, IvT, approach. It relies on various tools for problem-solving, e.g., modeling, simulation, virtual reality, data mining, artificial intelligence, rapid prototyping, high throughput chemistry, and high throughput screening. These technologies are becoming ubiquitous in the innovation process. The IvT approach has been used in the recent high profile projects, e.g., the design of the Millennium Bridge in London, reconstruction of the Leaning Tower of Pisa, the design, creation, and building of the Bilbao Guggenheim Museum, and solving London’s roadway congestion problem (Report 2004). Other innovation tolls include CREAX (Creax 2005), Visual Mind (Visual Mind 2005), and Pull Thinking (Pull Thinking 2005).

The above tools cover some aspects of the innovation space. Research is needed to identify gaps and explore other methodologies and tools enhancing innovation, e.g., creativity fostering tools. Yamamoto and Nakakoji (2005) described an interactive tool that impacts user's cognitive processes.

5. Conclusion

Increasing innovation awareness by the discovery of the underlying science is critical to corporations’ becoming progressive, competitive, and better prepared to handle future adversities. Innovation can fill the gap created by the shift in low-end manufacturing jobs and growing global market competitiveness. The paper outlined the need for the discovery of theories, processes, methodologies, and tools enhancing innovation. Some of the tools supporting innovation, e.g., genetic programming and data mining could be embedded in prototype software and integrated with the existing computational systems. Pattern discovery from data surrounding design, process, and service applications - and therefore data mining - and likely to become major solution approaches of the innovation cyber-infrastructure. The ramification and use of the existing theories (research is needed to formalize them), methodologies (e.g., group thinking, brainstorming), and innovation tools (e.g., TRIZ) needs be better understood, and new progressive models, methodologies, and tools should be developed.

Acknowledgement

Some of the ideas included in the paper have been discussed with my University of Iowa colleagues Clar Baldus, Linda Boyle, Patrick Butler, Yong Chen, LD Chen, Lawrence Fritts, Nagi Gebraeel, Jeffrey Marshall, Albert Ratner, Thomas Schnell, Lisa Troyer, and Stephen Morford from the US Army, Rock Island Arsenal, IL.
The term innovation science has been coined by the author of this paper at the NSF Workshop on Engineering Design in 2030.

References

Aha, D.W. (1992), Tolerating noisy, irrelevant and novel attributes in instance-based learning algorithms, *International Journal of Man-Machine Studies*, Vol. 36, No. 2, pp. 267-287.

Ahuja, G. (2000), Collaboration networks, structural holes, and innovation: A longitudinal study, *Administrative Science Quarterly*, Vol. 45, pp. 425-455.

Allen, K. (2003), *Bringing New Technology to Market*, Prentice Hall, Upper Saddle River, NJ.

Anderberg, M.R. (1973), *Cluster Analysis for Applications*, Academic Press, New York.

Ashworth, C.M. (1988), Structured systems analysis and design method, *Information and Software Technology*, Vol. 30, (April), pp. 153-163.

Auer, P., R. Holte and W. Maass (1995), Theory and application of agnostic PAC-learning with small decision trees, in A. Prieditis and S. Russell, Eds, *ECML-95: Proceedings of 8th European Conference on Machine Learning*, Springer, New York.

Barnes, B. (1982), The science-technology relationship: A model and a query, *Social Studies of Science*, Vol. 12, pp. 166-172.

Beekman, D. (1989), CIMOSA: Computer integrated manufacturing - open system architecture, *International Journal of Computer-Integrated Manufacturing*, Vol. 2, No. 2, pp. 94-105.

Bentley, P.J., and J.P. Wakefield, (1995), The table: An illustration of evolutionary design using genetic algorithms, *Proceedings of the 1st IEE/IEEE Conference on Genetic Algorithms in Engineering Systems: Innovations and Applications*, Sep 12-14.

Benzhaf, W., P. Nordin, R.E. Keller, and F.D. Francone (1998), *Genetic Programming; An Introduction*, Morgan Kaufmann, San Francisco, CA.

Browning, T.R. (2001), Applying the design structure matrix to system decomposition and integration problems: A review and new directions, *IEEE Transactions on Engineering Management*, Vol. 48, No. 3, pp. 292-306.

Caroll, J.M. and J. Olson (1987), *Mental Models in Human-Computer Interaction: Research Issues About the User of Software Knows*, National Academy Press, Washington, DC.

Clark, P. and R. Boswell (1989), The CN2 induction algorithm, *Machine Learning*, Vol. 3, No. 4, pp. 261-283.

Cohen, B.P., R.J. Kruse, and M. Anbar (1982), The social structure of scientific research teams, *Pacific Sociological Review*, Vol. 25, pp. 205-232.

Crawford, M. and A. Di Benedetto (2005), *New Products Management*, McGraw Hill, New York.

Daupert, D. (2005), *The Osborne-Parnes Creative Problem Solving Process Manual*, `http://www.ideastream.com/create`.

Coello, C.A.C. (1999), A comprehensive survey of evolutionary-based multiobjective optimization techniques, *Knowledge and Information Systems*, Vol. 1, No. 3, pp. 269-308.

Creax (2005), CREAX Innovation Suite 3.1, `http://www.creax.com/tools.htm`.

Dempster, A.P. (1977), Maximum likelihood from incomplete data via the EM algorithm, *Royal Journal of Statistical Society*, Series B, Vol. 39, pp. 1-38.

Domb, E. (2003), *Managing Creativity for Project Success*, The TRIZ Institute, http://www.triz-journal.com/whatistriz/index.htm.

Doumeingts, G., B. Vallespir, D. Darricar, and M. Roboam (1987), Design methodology for advanced manufacturing systems, *Computers in Industry*, Vol. 9, No. 4, pp. 271-296.

Eiben, A.E. and J.E. Smith (2003), *Introduction to Evolutionary Computing*, Springer, Heilderberg, Germany.

Engelbrecht, A.P. (2003), *Computational Intelligence: An Introduction*, John Wiley, New York.

Fonseca, C.M. and P.J. Fleming (1995), An overview of evolutionary algorithms in multiobjective optimization, *Evolutionary Computation*, Vol. 3, No. 1, pp. 1-16.

Friedman, J., Y. Yun and R. Kohavi (1996), Lazy decision trees, *Proceedings of the Thirteenth National Conference on Artificial Intelligence*, AAAI Press and MIT Press.

Grzymala-Busse, J. (1997), A new version of the rule induction system LERS, *Fundamenta Informaticae*, Vol. 31, pp. 27-39.

Han J. and M. Kamber (2001), *Data Mining: Concepts and Techniques*, Morgan Kaufmann, Palo Alto, CA.

Haribabu, E. (2000), Cognitive empathy in inter-disciplinary research: The contrasting attitudes of plan breeders and molecular biologists towards rice, *Journal of Biosciences*, Vol. 24, pp. 323-330.

Kantardzic, M. (2003), *Data Mining: Concepts Models, and Algorithms,* IEEE Press and John Wiley, New York.

Kohavi, R. (1995), Wrappers for Performance Enhancement and Oblivious Decision Graphs, Ph.D. Thesis, Computer Science Department, Stanford University, Stanford, CA.

Kostoff, R.N. (2002), Overcoming specialization, *BioScience*, Vol. 52, No. 10, pp. 937-941.

Kostoff, R.N. (2003), Role of technical literature in science and technology development and exploitation, *Journal of Information Science*, Vol. 29, No. 3, pp. 223-228.

Koza, J. (1992), *Genetic Programming*, MIT Press, Cambridge, MA.

Koza, J.R. (1994), *Genetic Programming II: Automatic Discovery of Reusable Programs*, MIT Press, Cambridge, MA.

Koza, J. (2001), www.genetic-programming.org.

Kusiak, A. (2000), *Computational Intelligence in Design and Manufacturing*, John Wiley, New York.

Kusiak, A. (2000a), Decomposition in data mining: An industrial case study, *IEEE Transactions on Electronics Packaging Manufacturing*, Vol. 23, No. 4, pp. 345-353.

Kusiak, A. (2001), Rough set theory: A data mining tool for semiconductor manufacturing, *IEEE Transactions on Electronics Packaging Manufacturing*, Vol. 24, No. 1, 2001, pp. 44-50.

Kusiak, A., N. Larson, and J. Wang (1994), Reengineering of design and manufacturing processes, *Computers and Industrial Engineering*, Vol. 26, No. 3, pp. 521-536 (http://www.icaen.uiowa.edu/%7Eankusiak/process-model.html).

Kusiak, A. and K. Park (1990), Concurrent engineering: Decomposition and scheduling of design activities, *International Journal of Production Research*, Vol. 28, No. 10, pp. 1883-1900.

Kusiak, A. and J. Wang (1993), Decomposition of the design process, *ASME Transactions: Journal of Mechanical Design*, Vol. 115, No. 4, pp. 687-695.

Mayer, R. J., T.P. Cullinane, P.S deWitte, W.B. Knappenberger, B. Perakath, and M.S. Wells (1992), Information Integration for Concurrent Engineering (IICE) IDEF-3 Process Description Capture Method Report, Armstrong Laboratory, AL-TR-1992-0057, Wright-Patterson AFB, Ohio.

Michalski, R.S. (1983). A theory and methodology of inductive learning, in R.S. Michalski, J.G. Carbonell, and T.M. Mitchell, Eds (1983), *Machine Learning: An Artificial Intelligence Approach*, Morgan Kaufmann, Los Altos, CA.

Michalski, R.S., I. Bratko, and M. Kubat, Eds (1998), *Machine Learning and Data Mining*, John Wiley, New York.

Michalski, R.S., I. Mozetic, J. Hong, and N. Lavrac (1986), The multi-purpose incremental learning system AQ15 and its testing application to three medical domains, *Proceedings of the 5th National Conference on Artificial Intelligence*, AAAI Press, Palo Alto, CA, pp. 1041-1045.

Michie, D., D.J. Spiegelhalter, and C.C. Taylor (1994), *Machine Learning, Neural, and Statistical Classification*, Ellis Horwood, New York.

NASA (2005), `http://ic.arc.nasa.gov/ic/projects/bayes-group/autoclass/`.

NIIR (2004), *Innovate America*, Council for Competitiveness, National Innovation Initiative Report.

Pawlak, Z. (1991), *Rough Sets: Theoretical Aspects of Reasoning About Data*, Kluwer, Boston, MA.

Piana, V. (2003), Innovation, `http://www.economicswebinstitute.org.`

Pinch, T.J. and W.E. Bijker (1990), The social construction of facts and artifacts: Or how the sociology of science and the sociology of technology might benefit each other, in Bijker W.E., T.P. Hughes, and T. Pinch, Eds (1990), *The Social Construction of Technological Systems*, MIT Press, Cambridge, MA, pp. 17-50.

Pull Thinking (2005), `http://www.pullthinking.com`.

Quinlan, J.R. (1993), *C4.5: Programs for Machine Learning*, Morgan Kaufmann, Los Altos, CA.

Quinlan, J.R. (1986), Induction of decision trees, *Machine Learning*, Vol. 1, No 1, pp. 81-106.

Report (2003), Cheskin and Fitch: Worldwide, *Fast, Focused & Fertile: The Innovation Evolution*.

Report (2004), Dodgeson, Gann and Salter, *Industrial Dynamics, Innovation and Development*, Elsinore, Denmark.

Schwery, A. and V.F. Raurich (2004), Supporting the technology-push of a discontinuous innovation in practice, *R&D Management*, Vol. 34, No. 5, pp. 539-552.

Shah, J. (2004), Engineering Design in 2030: An NSF Strategic Planning Workshop, `http://dal.asu.edu/engdesign/index.html`.

Siem, P. (1996), An Introduction to TRIZ: A Revolutionary New Product Development Tool, *Visions*, January.

Soule, T. and J.A. Foster (1998), Effects of code growth and parsimony pressure on populations in genetic programming, *Evolutionary Computation*, Vol. 6, No. 4, pp. 293-309.

Sternberg, R.J. (2005), Creativity or creativities?, *International Journal of Human-Computer Studies*, Vol. 63, No. 4-5, pp. 370-382.

Steward, D.V. (1981), The design structure system: A method for managing the design of complex systems, *IEEE Transactions on Engineering Management*, Vol. 28, pp. 71-74.

Tatikonda, M.V. and S.R. Rosenthal (2000), Successful execution of product development projects: Balancing firmness and flexibility in the innovation process, *Journal of Operations Management*, Vol. 18, pp. 401-425.

Tidd, J., J. Bessant, and K. Pavitt (2001), *Managing Innovation: Integrating Technological, Market and Organizational Change*, John Wiley, Chichester, UK.

TRIZ Journal (2005), `http://www.triz-journal.com`.

Troyer, L. (1995), Team Embeddedness: The Relations between Team Social Structures, Organization Social Structures, and Team Performance, Unpublished doctoral dissertation, Stanford University, Stanford, CA.

Troyer, L. (2004), Democracy in a bureaucracy: The legitimacy paradox of teamwork in organizations, in C. Johnson, Ed. (2004), *Research in the Sociology of Organizations: Legitimacy Processes in Organizations*, Elsevier, New York.

Troyer, L. (2005), Personal communication, Department of Sociology, The University of Iowa, Iowa City, IA.

UML (2005), Unified Modeling Language, `http://www.uml.org/`.

Urlich, K.T. and S.D. Eppinger (2000), *Product Design and Development*, McGraw-Hill, New York.

Utterback, J.M. and W.J. Abernathy (1975), A dynamic model of process and product innovation, *Omega - The International Journal of Management Science*, Vol. 3, No. 6, pp. 639-656.

Visual Mind (2005), `http://www.visual-mind.com`.

Wickens, G., S.E. Gordon, and Y. Liu (1998), *An Introduction to Human Factors Engineering*, Harper Collins, New York.

Yamamoto, Y. and K. Nakakoji (2005), Interaction design of tools for fostering creativity in the early stages of information design, *International Journal of Human-Computer Studies*, Vol. 63, No. 4-5, pp. 513-535.