Modeling Information Technology – A Pattern Approach for Enhancing Technology Intelligence Processes

Michael Schermann, Helmut Krcmar

Lehrstuhl für Wirtschaftsinformatik
Fakultät für Informatik
Technische Universität München
Boltzmannstraße 3
85748 Garching, Germany
michael.schermann@in.tum.de
krcmar@in.tum.de

Abstract: Nowadays, companies seeking to advance their information systems are facing an overwhelming variety of fast pacing innovations in the information technology sector. Many of these innovations could lead to new ways of extending existing competitive advantages or leveraging weaknesses respectively. Furthermore technological progress is characterized by shorter technology life cycles and by discontinuities. The challenge for developing information systems that contribute to a company's competitive advantage is to gather intelligence information of technological progress, evaluate possible strategic impacts of that information and prepare for decision making. We argue that technologies may be seen as a form of knowledge and therefore propose a pattern based modeling language to capture and model information technologies. We demonstrate the applicability of the modeling language on the example of web service related technologies.

1 Introduction

As one of the foundations of information society, IT determines the business competency and becomes a critical success factor of IT management [ZB03], [Sz81]. Strategic intelligence information about the technological progress and new trends in the IT market as well as in scientific communities is therefore determining future success in the same way as information about customers or competitor's actions [Ts98]. Companies are therefore facing the challenges of identifying relevant new IT trends and of evaluating the impacts of these trends on their way of doing business. Such tasks are usually summarized by the concept of technology intelligence e.g. [Zw99]. Generally technology intelligence begins with identifying relevant information about information technology. Despite the importance, the concept and structure of technology is not fully understood yet. Managing information technology requires identifying and capturing important aspects of technologies. Therefore, the first question we address in the course of this paper is: What is the main structure of technologies?

In architecture and software engineering, patterns and pattern languages have been proven tools of capturing knowledge, e.g. design experience (Alexander 1979, Schumacher 2003). We will argue that technologies may be defined as problem oriented knowledge and thus may be captured using patterns. However, patterns are usually presented in natural language (e.g. Alexander 1979). Therefore we present a modeling language for modeling pattern languages to support documenting existing knowledge about technologies. Furthermore we analyze pattern construction methods originated in the field of software engineering for their suitability in technology intelligence processes. Hence, the second question we address in this paper is: What are the elements and process of modeling information technologies? We will demonstrate the applicability of the modeling language by analyzing technologies in the vicinity of web services.

The remainder of this paper is structured as follows: In the next section 2 we will briefly discuss the process of technology intelligence. Furthermore we will review the concept of information technology and propose a knowledge based definition thereof. In the following section 3 we will introduce the idea of patterns and pattern languages and discuss common methods for gathering and developing patterns. Furthermore we discuss their applicability in technology intelligence processes. The section closes with presenting a meta model of a modeling language for IT patterns. Then we apply the modeling language by creating a technology pattern language for service-oriented architectures. The paper concludes with a short summary of results achieved and an outlook on avenues for future research.

2 Core Concepts

2.1 IT Intelligence

The tasks of identifying risks and opportunities arising from technological progress, evaluating possible impacts, and deducing recommendations for actions are commonly summarized by the concept of technology intelligence [Zw99]. Thus information Technology (IT) intelligence as a task of the strategic information management may be characterized as identifying, analyzing, and evaluating innovations in the domain of information technology (cf. [Kr05]).

The importance of IT intelligence can be justified as follows: deployed information technologies determine the technological potential of information systems and hence determine the contribution of information systems to an overall success. Furthermore information technologies shape the technological environment of a company. For instance, information technologies such as web services could enable companies to outsource certain functions of their information systems. Thus companies can focus on strategic relevant functions of their information systems [EKT02]. Therefore strategic effective deployment of innovative information technology can be seen as a critical success factor [ZB92].

Bürgel et al. find that the task of technology intelligence is to provide answers to the following questions [BRZ02]:

- What are the main trends in information technology?
- Which impact on existing information systems can be deduced?
- Which recommendations can be derived for the technological strategy?

According to these questions, the technology intelligence process can be differentiated into three main phases (see Figure 1). Technology intelligence begins with identifying technological innovations at an earliest state possible by detecting so called weak signals (Ansoff 1975). Scanning and monitoring the technological environment results in a diagnosis of possibly relevant technological trends.

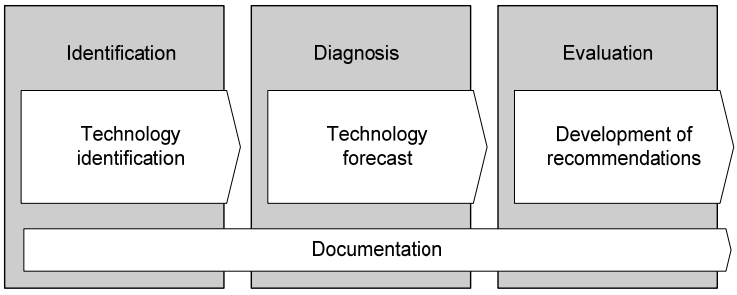

Figure 1: The technology intelligence process according to [Ze03]

The main activity in the second phase is forecasting development of trends and technologies. Following this, recommendations have to be derived for adapting the information infrastructure accordingly. An overall task of technology intelligence is documentation [Ze03]. In this paper we focus on the documentation part of the technology identification phase.

2.2 Weak signals as the theoretical nucleus of technology intelligence

A theory often discussed in the context of Business Intelligence is the concept of weak signals, proposed by [An75]: companies usually use only a small amount of available information for preparing decisions. Furthermore information gets adopted and diffused in societies over time [KM83]. Thus discontinuous situations, e.g. the dawn of a new technological paradigm, are not discontinuous at all, but develop over the time. They can be identified so called weak signals.

A **weak signal** can be defined as a certain set of data, which can be interpreted ambiguously only. As time continues, the data from an evolving situation becomes clearer, but a possibly needed response time gets shorter. Hence available information has to determine potential actions, not decision processes determines the potential set of requires information. As depicted in Figure 2 [An75] develops five strategies based on the characteristics of the available information, which allow staged actions. An often met criticism linked with the concept of weak signals is an absent operationalization. In the

context of technology intelligence, such operationalization can be derived from innovation processes (e.g. [Ro95]), as depicted in Figure 2.

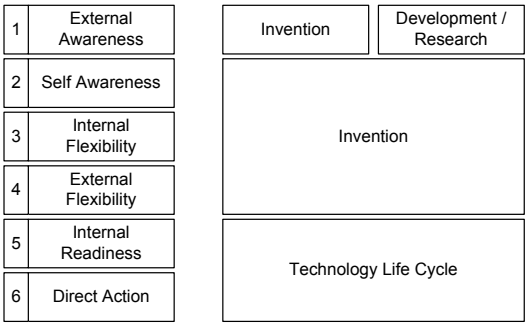

Figure 2: Comparison of Ansoffs weak signal strategies and an ideal innovation process [An75], [Br80]

The comparison of an ideal innovation process and Ansoffs strategy stages allows a translation of the concept of weak signals to situations that occur in ithe progress of technological progress. Therefore is it possible to classify gathered information and derive potential action strategies from it.

2.3 Information Technology as Knowledge

Following Tondl or Bunge, the term ‘technology’ can be defined as a specific form of knowledge which allows the human manipulation of the natural environment to a specific purpose [To74], [Bu74]. Thus technologies can be characterized as problem solving knowledge entities [To74]. Furthermore technologies can rarely be described or applied without referencing other technologies. Hence references between technologies are an important characteristic of technologies [Za95].

In the context of information management, the purpose of applying new information technologies is to counteract identified weaknesses in the technological infrastructure or extend and hence increase the potential of success of the infrastructure. Gathering information about technological progress requires a systematic method to manage complexity facing shorter development cycles, faster technological progress and an overwhelming variety of information technologies. As technology can be seen as form of knowledge, we propose patterns as a suitable way of modeling and thus systematizing technologies.

3 A Pattern Language for Information Technology

3.1 Patterns and Pattern Languages

By virtue of apparently unlimited number of information technologies on one hand and the likewise unlimited possibilities of combinations on the other hand the main objective of documenting information technologies seems to be managing complexity [St94]. As we discussed above, the documentation of gathered information is an overall task in technology intelligence. Hence it is quite surprising that only few options have been proposed e.g. [KM93]. Based on our definition of technology as kind of knowledge we discuss a pattern-based approach for modeling information technology.

The idea of **patterns** has been originally developed in architecture by Alexander et al. and can be summarized as the combination of a solution to architectural problems on the base of a general framework [Al79]. In line with cognitive research a pattern can be defined as a representation of knowledge [Bo01]. In the field of software engineering, Gamma et al. transferred the pattern idea to programming and design problems [Ga95].

A pattern generally comprises the following elements [Sc03]: the **context** comprises causes which lead to the problem described in a pattern and the conditions under which the problem occurs. The context should support acquiring the relevance of a pattern [Bu98]. The **problem** is to be described by explaining contradictions causing the problem in the context of the pattern. These aspects of the pattern problem section are often called forces [Bu98]. The next section of a pattern explains the proposed **solution** by dissolving forces described before. An illustration of possible side effects is given as well [Bu98]. The closing section of a pattern is composed of **references** to related patterns [Sc03].

In sum a pattern represents a complex structure of knowledge from an application-oriented perspective. The goal of patterns is to explicate experiences and established expert knowledge [Bo01]. As patterns are rarely used independently, Alexander broadens the pattern idea to a system of interrelated patterns that he called **pattern language** [Al79]**.** The semantic power of such pattern languages is determined by the references between patterns, which consequently allow capturing solutions for more complex problems [Sc03].

3.2 Definition of an information technology pattern language

In the following, we introduce the modeling elements that represent information technologies as IT patterns. First, the components of patterns will be explained, followed by a description of possible references between IT patterns. The meta model describing the modeling language is depicted in Figure 3.

The *context* of an IT pattern contains the general conditions and depicts the business environment of a certain technology. The goal of the context is to clarify the problem

domain of an information technology. The *problem* describes given deficiencies in the context by showing forces in the problem domain, e.g. existing technologies. For example, it can call attention to a lack of an aggregated information supply as a result of monolithic application systems. The *solution* of the pattern is unfolding an architectural appliance of technology to solve the given problem. The solution should be of a reference character that means it should depict the technological aspect in a general way [Bo01]. The concept of *strategic option* refers to the strategic alternatives proposed in [An75] (see section 2.2).

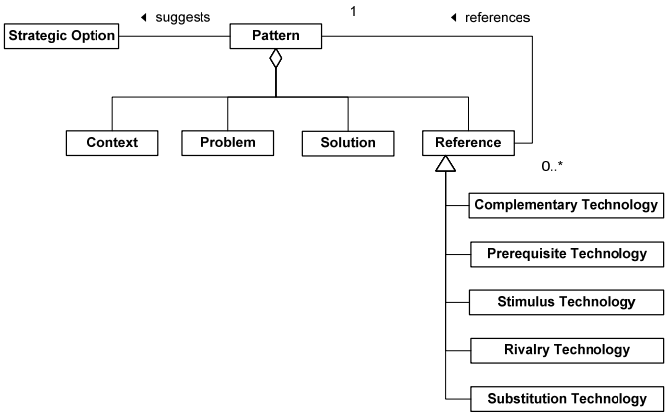

Figure 3: Basic Meta Model of the Modeling Language for IT Pattern Languages in UML [OMG03]

The references constitute an important element of an IT pattern as references describe side effects that have to be handled by another technology. For instance, the information technology *Online Analytical Processing (OLAP)* assumes the appliance of the information technology *Data Warehouse*. To evaluate technological progress or impacts on an IT infrastructure it is necessary to analyze relationships of a given set of technologies. Hence we differentiate the following reference types cf. [Za95], cf. [Pf89]: **Complementary technologies** entail the appliance of each other. In contrast a **prerequisite technology** is required for applying another technology. A **stimulus technology** describes a relationship where the progress of one technology aides the progress of another. Unlike this, the progress of a **rivalry technology** oppresses the progress of related technologies, as both technologies allow the solving of the same problem using different technical solutions. The strong form of a rivalry relationship is the **substitution technology**, which supersedes another technology by applying a better economical or technological alternative [Ew89].

This specification of an IT pattern language allows gathering information about information technologies systematically and analyzes their relationships in order to gain relevant information for the technology intelligence process.

3.3 The Process of Gathering IT Patterns

In the following we explain two approaches to support the gathering and construction of IT patterns. As a result of the specificity of a given company setting, we conclude that the technology intelligence process has to be developed dependent on this setting and the conducting experts cf. [BRZ02]. That is to say that the pattern development process is highly subjective, since it reflects the conclusions of the developing IT expert. Therefore a pattern holds the possibility of entailing the experts' bias towards the described technology. Schumacher et al. summarize two quality enhancing processes to advance the inter-subjectivity of given patterns [SRM03]:

Shepherding can be characterized as review process where a pattern expert supports the originator of patterns in improving and advancing the patterns. The reviewing expert takes a passive role only, so his responsibility is to propose improvements. Therefore the goal of shepherding can be seen in supporting consistent, clearly formulated and coherent patterns [SRM03].

The **Writer's Workshop** supports the writing of patterns as it forms a moderated body of experts with a passive attendance of the pattern's originator. The originators passivity is supposed to avoid justification by the originator and to support the acceptance of the workshop's results as suggestions for improvement. The task of the workshop is therefore to evaluate the developed patterns by pursuing the following process: First the patterns of concern are introduced to the experts by two of them assigned specifically. In doing so, the originator can assert whether the patterns where written comprehensibly to the public. Following this, the patterns will be analyzed and evaluated by the body of experts. Besides the formal aspects, the experts especially focus on the relationships between patterns. At the end of the workshop the originator becomes active in summing up the suggestions and making sure he has understood the experts correctly [SRM03].

We have to remark, that both, Shepherding and the Writer's Workshop have been developed to support the pattern development process in software engineering. The goal of a pattern in software engineering is to provide a solution for a recurring, but mainly time invariant problem. In a technology intelligence process these two approaches are used to monitor a technology and filter irrelevant information respectively. Therefore the focus of both has to be put on the novelty of an underlying technology. For example, a goal of a Writer's Workshop can be to determine if a certain IT pattern is basically the transfer of an existing solution to another context. Furthermore both approaches support an inter-subjective development of IT pattern languages by focusing on the references of the IT patterns. In doing so, it is possible to identify illusory innovations or technology assimilations by embedding new IT patterns in an existing IT pattern language. In sum the goal of developing an IT pattern language is to construct and maintain a best possible inter-subjective model of the information technological environment.

3.4 Critical Analysis of the IT Pattern Approach

IT patterns facilitate a uniformed way of modeling a technology environment which supports comparability of technologies and their characteristics respectively. Thus representing information technologies by IT pattern languages allows managing technological complexity by providing a semiformal approach of documenting technologies. By applying construction methods such as shepherding a consistent and commonly shared understanding of information technologies may be gained. Thus, an important characteristic of an IT pattern language can be seen in supporting communicational processes about information technology. Following the basic idea of patterns, IT patterns facilitate the transfer of expert knowledge to a technology intelligence process.

As we argued above, IT patterns allow the representation of new information technologies, which means in turn that some parts or even whole elements of IT patterns are still uncertain or unknown respectively. Also we remark, that in contrary to patterns in software engineering, IT patterns show a much higher number of revisions to keep track of the corresponding technological progress. Therefore it is necessary to document changes made to the elements of a pattern as well. In doing so, this process of versioning allows to identify possible invariant parts of an IT pattern likewise.

In software engineering patterns are used to represent expert knowledge in form of well-established solutions with the goal of enhancing software development processes. Thus Schumacher et al. conclude that patterns do not describe poorly conceived ideas or inventions, which casts our IT pattern approach into doubt [SRM03]. We have chosen the structure of patterns as base of our modeling language because of the ability to represent knowledge. We argue that there are no structural differences between a pattern representing experience knowledge and a pattern representing problem solving knowledge as we defined technology [To74]. Hence, the pattern approach is applicable to document the progress and relationships of information technologies. We basically use the pattern idea in a different context for different purposes.

Another problem in gathering IT patterns and constructing a usable pattern language is to identify and model technologies that possess a similar degree of abstraction to ensure for instance comparability of patterns. As we have discussed above the pattern construction process is highly subjective. Thus the technological granularity or abstraction of the patterns is subjective as well. Furthermore in section 3.3 we have introduced two methods to enhance inter-subjectivity in the process of constructing IT pattern. Established information system models such as [Za87] can be used to ensure similar degrees of abstraction when construction a pattern language.

4 Example: Service-Oriented Architecture

4.1 Introduction to Service-Oriented Architectures

The basic idea behind service-oriented architectures (SOA) is the concept of service. A **service** can be defined as a component which fulfils a specified and published function. Instead of monolithic applications a service-oriented application comprises a lot of loosely coupled services. Hence, a SOA allows easier maintenance and promises faster and less error-prone changes to information systems [KL04]. When developing an application, service-oriented architectures enhance reuse of services and even the incorporation of externally developed services. While these goals are not new, one thing makes SOA special: it is based on a commonly accepted set of standardized protocols called web services. Web services are a technology which allows encapsulating certain functionality and making them available over common Internet protocols. In addition to the opportunity of enabling an easier integration of applications, service-oriented architectures can be used to implement business processes with the advantages of loose coupling, even across company borders [EKT02]. In sum service-oriented architectures claim to enable modular business processes, that can be easily changed, outsourced and reintegrated as service [Wi02].

4.2 An IT pattern language for service-oriented architectures

Figure 4 shows five information technologies that are discussed in the context of service-oriented architectures. Each technology is modeled as an IT pattern and is displaying a short description of its context, the problem and solution. The references shown illustrate the relationships between the technologies. While web services allow the encapsulation of functionality and thus allow the integration of a heterogeneous applications they do not provide a way of implementing processes. Such functionality is provided by the Business Process Execution Language for Web Services (BPEL4WS).

Both technologies are complementary because both are required to implement business processes. Web services can be seen as rivalry technology to Common Object Request Broker Architecture (CORBA) as it solves a similar problem. BPEL4WS in turn can be seen as a substitution technology to XLANG and WSFL (Web Service Flow Language) as it provides almost same functionality of both technologies and has been developed to replace them. Likewise, XLANG and WSFL have a rival relationship as both allow implementing business processes, but follow different approaches.

The number in the upper right corner of each pattern indicates applicable strategic options as discussed in [An75] (see section 2.2). For instance, BPEL4WS is still in the process of definition and can be seen as a proposal for the implementation of business processes. BPEL4WS has not yet achieved an acceptance as for instance Web services. Therefore we conclude there is a lot of uncertainty included when using BPEL4WS or any other similar approach while pursuing the goal of service-oriented architectures.

Therefore we conclude that at this point in time it is possible to integrate applications using web services but implementing business processes using BPEL4WS is still in the innovation phase. For instance a middle-sized company pursuing a service-oriented architecture may thus begin with integrating legacy applications and even use technologies that allow implementing business processes but has to be aware of changes in the relevant technologies (strategic option: internal flexibility). From the perspective of technology intelligence the development of BPEL4WS should be closely monitored.

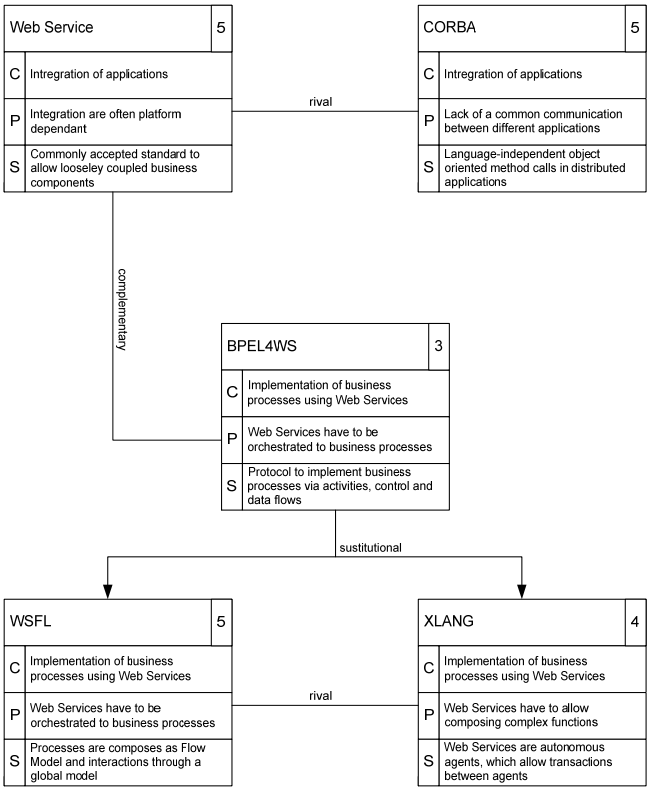

Figure 4: Information technologies that are relevant in the domain of service-oriented architectures

Despite the simplified example we have demonstrated that the IT pattern approach allows the systematic interpretation of a set of related technologies and provides useful information for further technology intelligence pursuits.

5 Summary and Outlook

Although we have presented research in progress the IT pattern approach is a possible way to model information technologies and their relationships. In doing so, we support the documentation process of the technology identification phase by providing a

manageable way to documenting relevant elements of technologies and support the interpretation of possible impacts on a company's information system. The proposed IT pattern approach allows a systematic gathering of information about information technologies and supports communication and cooperation processes in the technology intelligence process.

Further research could address the following issues:

- Empirical validation of proposed benefit of applying patterns for systematizing technology documentation
- Modeling technological aspects of existing application systems and infrastructure.
- Using patterns for easing communication and cooperation in technology intelligence processes.
- Combine patterns on different levels such as business processes, application systems, and technology.
- Transferring the idea of modeling technology as patterns to other areas of technology management.

Overall, modeling information technology based on our pattern approach allows analyzing technologies in a systematic way.

6 References

[ZB03] Zarnekow, R. and W. Brenner, *Konzepte für ein produktorientiertes Informationsmanagement*, W. Uhr, W. Esswein, and E. Schoop, Editors. 2003, Physica-Verlag: Heidelberg. p. 735-753.

[Sz81] Szyperski, N., *Geplante Antwort der Unternehmung auf den informations- und kommunikationstechnischen Wandel*, in *Organisation, Planung, Informationssysteme*, E. Frese, P. Schmitz, and N. Szyperski, Editors. 1981, Carl Ernst Poeschel Verlag: Stuttgart. p. 177-195.

[Ts98] Tschirky, H., *Technologie-Management: Schließung der Lücke zwischen Management-Theorie und Technologie-Realität*, in *Technologie-Management: Idee und Praxis*, H. Tschirky and S. Koruna, Editors. 1998, Verlag Industrielle Organisation: Zürich. p. 1-32.

[Zw99] Zweck, A., *Technologiefrüherkennung - Ein Instrument zwischen Technologiefolgenabschätzung und Technologiemanagement*, in *Handbuch Technologiefolgenabschätzung*, S. Bröchler, G. Simonis, and K. Sundermann, Editors. 1999, Edition Sigma: Berlin. p. 155-164.

[Kr05] Krcmar, H., *Informationsmanagement*. 4., überarb. und erw. Aufl. ed. 2005, Berlin: Springer. XIX, 574 S.

[EKT02] Engel, A., A. Koschel, and R. Tritsch, *J2EE kompakt - Enterprise Java: Konzepte und Umfeld*. 2002, Heidelberg: Spektrum Akademischer Verlag.

[ZB92] Zahn, E. and F. Braun, *Identifikation und Bewertung zukünftiger Techniktrends - Erkenntnisstand im Rahmen der strategischen Unternehmensführung*, in *Technologiefrühaufklärung: Identifikation und Bewertung von Ansätzen zukünftiger Technologien*, VDI-Technologiezentrum Physikalische Technologien, Editor. 1992, Schäffer-Poeschel Verlag: Stuttgart. p. 3-15.

[BRZ02] Bürgel, H.D., G. Reger, and R. Ackel-Zakour, *Technologie-Früherkennung in multinationalen Unternehmen: Ergebnisse einer empirischen Studie*, M.G. Möhrle and R. Isenmann, Editors. 2002, Springer-Verlag: Berlin. p. 19-45.

[Ze03] Zeller, A., *Technologiefrühaufklärung mit Data Mining: Informationsprozessorientierter Ansatz zur Identifikation schwacher Signale*. 2003, Wiesbaden: Deutscher Universitäts-Verlag.

[An75] Ansoff, H.I., *Management Strategic Surprise by Response to Weak Signals.* California Management Review, 1975. **18**(2): p. 21-33.

[KM83] Krampe, G. and G. Müller, *Diffusionsfunktion als theoretisches und praktisches Konzept für Strategische Frühaufklärung*, in *Bausteine eines strategischen Managements: Dialoge zwischen Wissenschaft und Praxis*, W. Kirsch and P. Roventa, Editors. 1983, Walter de Gruyter & Co.: Berlin. p. 283-304.

[Ro95] Rogers, E.M., *Diffusion of Innovations*. 4 ed. 1995, New York: The Free Press.

[Br80] Bright, J.R., *Practical Technology Forecasting: Concepts and Exercises*. 1980, Austin, Texas: The Industrial Management Center.

[To74] Tondl, L., *Technology as Applied Science*, in *Contributions to a Philosophy of Technology*, F. Rapp, Editor. 1974, D. Reidel Publishing Company: Dordrecht, Holland. p. 1-18.

[Bu74] Bunge, M., *Technology as Applied Science*, in *Contributions to a Philosophy of Technology*, F. Rapp, Editor. 1974, D. Reidel Publishing Company: Dordrecht, Holland. p. 19-39.

[Za95] Zahn, E., *Gegenstand und Zweck des Technologiemanagements*, E. Zahn, Editor. 1995, Schäffer-Poeschel Verlag: Stuttgart. p. 3-32.

[St94] Steinbock, H.-J., *Potentiale der Informationstechnik: state of the art und Trends aus Anwendungssicht*. 1994, Stuttgart: B. G. Teubner.

[KM93] Krystek, U. and G. Müller-Stewens, *Frühaufklärung für Unternehmen: Identifikation und Handhabung zukünftiger Chancen und Bedrohungen*. 1993, Stuttgart: Schäffer-Poeschel Verlag.

[Al79] Alexander, C., *The timeless way of building*. 1979, New York: Oxford University Press.

[Bo01] Borchers, J., *A Pattern Approach to Interaction Design*. 2001, Chichester: John Wiley & Sons.

[Ga95] Gamma, E., et al., *Design Patterns: Elements of Reusable Object-Oriented Software*. 1995, Reading, Massachusetts: Addison-Wesley Publishing Company.

[Sc03] Schumacher, M., *Security engineering with patterns: origins, theoretical models, and new applications*. Lecture notes in computer science; 2754. 2003, Berlin: Springer.

[Bu98] Buschmann, F., et al., *Pattern-orientierte Software-Architektur: Ein Pattern-System*. 1998, Bonn: Addison-Wesley-Longman Verlag.

[OMG03] Object Management Group. *OMG Unified Modeling Language Specification Version 1.5*. 2003 [cited 2005 05-01-2005]; Available from: http://www.omg.org/cgi-bin/apps/doc?formal/03-03-01.pdf.

[Pf89] Pfeiffer, W., et al., *Technologie-Portfolio zum Management strategischer Zukunftsgeschäftsfelder*. 1989, Göttingen: Vandenhoeck & Ruprecht.

[Ew89] Ewald, A., *Organisation des Strategischen Technologie-Managements: Stufenkonzept zur Implementierung einer integrierten Technologie- und Marktplanung*. 1989, Berlin: Erich Schmidt Verlag.

[SRM03] Schumacher, M., U. Rödig, and M.-L. Moschgath, *Hacker Contest - Sicherheitsprobleme, Lösungen, Beispiele*. 2003, Berlin: Springer Verlag.

[Za87] Zachman, J.A., *A Framework for Information Systems Architecture.* IBM Systems Journal, 1987. **26**(3).

[KL04] Kossmann, D. and F. Leymann, *Web Services.* Informatik Spektrum, 2004. **27**(2): p. 117-128.

[Wi02] Wiedemann, M., *Dichtung und Wahrheit - Web Services im Collaborative Commerce.* Information Management & Consulting, 2002. **17**(3): p. 57-60.