Development of a Generic Tool for the Application of Maturity Models - Results from a Design Science Approach

Nico Krivograd
Institute für Wirtschaftsinformatik (IWi) within Deutsches Forschungszentrum für Künstliche Intelligenz (DFKI)
Saarbrücken, D-66123, Germany
Email: Nico.Krivograd@iwi.dfki.de

Dr. Peter Fettke
Institute für Wirtschaftsinformatik (IWi) within Deutsches Forschungszentrum für Künstliche Intelligenz (DFKI)
Saarbrücken, D-66123, Germany
Email: Peter.Fettke@iwi.dfki.de

Abstract
The determination of the maturity level of an object or an activity is an expensive and complex challenge. Tools for determining the maturity level offer a way to reduce costs and complexity. Such tools can be used by persons, even if they have not received sophisticated training for the applied maturity model. As a result, high training costs can be avoided. Furthermore, the costs of a maturity assessment are minimized because the tools automatically perform the analysis. This paper develops a list of requirements which a tool must fulfill for the application of maturity models and presents a possible implementation of such a tool. As the implemented tool can handle several maturity models, we refer to it as a generic tool.

1. Introduction
There is a variety of different maturity models today and their number will increase in the future [1]. For example, some researchers from Queensland University of Technology are currently developing a Business Process Management Maturity model to assess the BPM success, i.e. the actual process performance [10].

A major challenge of the various models is that they require manual usage and evaluation which is difficult and very expensive. Therefore, tools are needed which support and facilitate the usage of maturity models. Such a tool should support mainly two aspects. First, it should help a user who has only basic knowledge of a certain maturity model to quickly and easily determine the maturity level of an object or an activity and thus detect weaknesses and potential improvements. As effect, the analysis of the maturity level can be done by the persons directly involved in the process, such as the internal employees of an organization. The engagement of external experts is not necessary with the consequence of saving the experts’ fees. Second, such a tool should automate processes and analyses to reduce the time effort of a maturity assessment. To sum it up, the tool should enable a simple, convenient and automated determination of the maturity level of an object or an activity.

In this study, the requirements which a tool for the application of maturity models should meet are developed as basis for the further research. Then, a tool example is developed as a prototype. We decided to follow a Design Science approach to achieve that our tool can enable organizations to address important information-related tasks [5]. Design Science approaches were already successfully proven to develop requirements for the development of maturity models [1].

The paper is structured as follows: Section 2 gives an overview of the state of the art. In section 3, the requirements are developed which a tool must fulfill for the application of maturity models. In section 4, the design and the application of the tool prototype are described. Section 6 summarizes the results of the study.

2. State of the Art
During the last years, several manufacturers have released so-called business process management tools to support process management. These tools pursue a number of approaches and concepts, covering generally the management areas important in this context: objectives, strategy, documentation, analysis, optimization and automation. Maturity models can provide help in the design of processes. A survey of the Frauenhofer Institute for Industrial Engineering shows, however, that maturity models are rarely included in the standard versions of these tools. Among the
Table 1. Software products that support maturity determination following [11]

<table>
<thead>
<tr>
<th>Manufacturer</th>
<th>Products</th>
<th>Maturity Model BPM</th>
<th>Maturity Model CMM</th>
<th>Maturity Model SPICE</th>
<th>Maturity Model ISO</th>
<th>Maturity Model PMM</th>
</tr>
</thead>
<tbody>
<tr>
<td>binner-IMS</td>
<td>Sycat</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
</tr>
<tr>
<td>BOC</td>
<td>ADONIS</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
</tr>
<tr>
<td>Dr. Lürzer</td>
<td>Promol.NET</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
</tr>
<tr>
<td>EMPRISE</td>
<td>BONAPART</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
</tr>
<tr>
<td>IPK</td>
<td>MO²GO</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
</tr>
<tr>
<td>IDS Scheer</td>
<td>ARIS</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
</tr>
<tr>
<td>IMG / S&amp;T</td>
<td>SemTalk/ Promet@work</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
</tr>
<tr>
<td>MID</td>
<td>Innovator</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
</tr>
<tr>
<td>pulinco</td>
<td>TopEASE</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
</tr>
<tr>
<td>VICon</td>
<td>VIFlow</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
</tr>
</tbody>
</table>

- ○ Maturity determination rudimentarily supported
- ☐ Maturity determination scheduled

examined tools, only the tool TopEASE by pulinco provides support for several maturity models. There are four additional tools which provide support for one chosen maturity model each. For all other tools, a maturity assessment is only offered on a separate request to the tool manufacturer (see Table 1) [11]. The Frauenhofer Institute for Industrial Engineering has also examined the level of support. The existence of an evaluation scheme, the representation of the maturity level and the documentation of the maturity assessment were selected as criteria for the degree of support. The examined tools ADONIS by BOC, ARIS by IDS Scheer and TopEASE by pulinco offer all rating scheme for the BPM maturity model. The representation of maturity levels is different: TopEASE follows the requirements of the model. ADONIS uses special model types. In ARIS, however, the representation is implemented customer-specific. The documentation of the maturity level is supported by TopEASE with several libraries, reports and a HTML portal. ADONIS documents the level of maturity with bar and spider diagrams. ARIS requires customized implementations.

The investigation of business process management tools by the Frauenhofer Institute draws the conclusion that their implementation of maturity models is "customer and mainly project specific" [7]. Outside the process view, no tools are known to support maturity models. Therefore, we come to the conclusion that there is a significant gap in tools for maturity models that should be addressed.

The first step in the development of a tool for the application of maturity models is the clarification of the requirements which such a tool should fulfill. Our research did not reveal relevant documents which adequately examine and describe these requirements. Therefore, we included the clarification of the requirements as first step in our research based on general requirements of software products.

3. Selected Requirements

The first step in our research was the clarification of the requirements of a tool for the application of maturity models as these were not yet examined in the available literature. The objectives that the tool should achieve were our starting point: First, the tool should be usable by a person that has only little knowledge about the maturity model. Second, it should automate processes and analyses.

We started with the question on how a person can do a maturity assessment without detailed knowledge of the used maturity model. To solve that question, we developed a structure to represent the maturity model which leads the user through the complete assessment of the model’s assessment criteria. Our research revealed that all examined models can be represented by a catalogue of questions which represent the model’s assessment criteria and by a choice of answer options which represents the different degrees to which an assessment criteria is fulfilled. Based on this representation of a maturity model, the tool can automatically calculate the result of the maturity assessment. As a result, the person using the tool is led through the complete assessment to the result. A specific knowledge is not necessary.
Table 2. Requirements for the generic tool

<table>
<thead>
<tr>
<th>Requirement</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Genericity</strong></td>
<td>The tool's life cycle is adapted by introducing a separate configuration-time for the implementation of specific customer demands.</td>
</tr>
<tr>
<td>Support of multiple maturity models</td>
<td>The tool is able to work with various maturity models.</td>
</tr>
<tr>
<td>Support of different scale levels</td>
<td>The tool is able to work with different scale levels.</td>
</tr>
<tr>
<td>Extensibility</td>
<td>The tool is designed in such a way that it can be easily adapted and extended to work with more maturity models.</td>
</tr>
<tr>
<td>Connectivity</td>
<td>The tool has an interface to connect to external applications.</td>
</tr>
<tr>
<td>Simplicity</td>
<td>The tool is able to quickly and easily support regular assessments.</td>
</tr>
<tr>
<td>Ease of use</td>
<td>The tool is designed in such a way that users with only basic training can intuitively perform an assessment.</td>
</tr>
<tr>
<td><strong>Functional</strong></td>
<td></td>
</tr>
<tr>
<td>Create and delete user and client</td>
<td>The tool is able to manage multiple clients and users.</td>
</tr>
<tr>
<td>Create, edit and delete objectives</td>
<td>The tool is designed in such a way that changes of the questions can be done fast and easily.</td>
</tr>
<tr>
<td>Create, edit and delete answer options</td>
<td>The tool is designed in such a way that changes of the answer options that can be done fast and easily.</td>
</tr>
<tr>
<td>Create, edit and delete model results</td>
<td>The tool is designed in such a way that changes of the model results that can be done fast and easily.</td>
</tr>
<tr>
<td>Weight answer options</td>
<td>The tool is able to weight different answer options independently.</td>
</tr>
<tr>
<td>Evaluate an assessment automatically</td>
<td>The tool is able to automatically determine the maturity level on the basis of the responses.</td>
</tr>
<tr>
<td>Generate reports</td>
<td>The tool is able to generate result reports on the basis of the assessments.</td>
</tr>
<tr>
<td>Compare assessments</td>
<td>The tool supports the automatic comparisons of assessments from different time points.</td>
</tr>
</tbody>
</table>

With regard to the second demand, we continued to identify the functions that the tool should offer at least to enable an appropriate handling. According to our understanding, the tool needs to offer some administrative functions like creating or deleting a user. Then, the tool needs to offer functions to create a new model with all its elements or to adapt the model elements. Finally, the tool needs to offer some functions which support the usage of the maturity assessment, like e.g. a result report. We came to the conclusion that the tool must be generic, i.e. it should be able to support different maturity models. As a consequence, it is of high importance that the tool reflects the different model structures and evaluation processes and that the tool can handle different types of scales. Furthermore, we see a strong need that the tool can be connected with other applications to exchange data and to mutually leverage the others’ functionalities.

We clustered the above described requirements in non-functional and functional ones. Table 2 lists the identified requirements. The functional requirements are quite easy to understand so that we abandon on their detailed description here. The first four non-functional requirements seem to be the most interesting ones for us as we believe that they make the difference between innovation and another idle stuff. We are convinced that these four requirements decide about the success and the sustainability of the tool. Therefore, these requirements are described in detail below.

**Genericity:** To meet the individual needs of organizations, it is necessary to have an appropriate configuration phase for the generic maturity tool. Therefore, the life cycle of a standard tool is modified accordingly. For a standard tool, no or only a few configuration is required. Therefore, the configuration is carried out usually at the end of the build-time or at the beginning of the runtime. For the generic maturity model tool, a special configuration-time is scheduled which is placed between the build-time and the runtime (see Figure 1).

The “build-time in the strict sense” reflects the pure development of the tool. During this time, the basic structure of the tool is developed and all generic groundwork is laid so that the tool can map various maturity models. The configuration-time is used to establish the connections with a process repository, to adapt existing maturity models to the specific client or to create new maturity models. The “runtime in the strict sense” reflects the actual application of the tool. In the runtime, only minimal configurations occur such as creating a new user or changing the weights of some questions.

The database structure of the tool is designed in a way so that it is reusable and has not to be redesigned for each individual client's needs. Because of this generic skeleton which was developed in the “build-time in the strict sense”, the tool is referred as generic.

**Support for multiple maturity models:** A company often gets requirements to fulfill certain levels of different maturity models by different customers. An example could be an electronics
Life cycle of a standard tool:

<table>
<thead>
<tr>
<th>Build-time</th>
<th>Runtime</th>
</tr>
</thead>
</table>

Modified life cycle of the generic maturity tool:

<table>
<thead>
<tr>
<th>Build-time in the strict sense</th>
<th>Configuration-time</th>
<th>Runtime in the strict sense</th>
</tr>
</thead>
</table>

Figure 1. Standard life cycle and modified life cycle of the tool

manufacturer which is supplier for a car manufacturer and an air carrier. The car manufacturer requires at least SPICE Level 3 and the air carrier asks for at least CMMI Level 4. To handle this situation, the tool should be able to work with several different maturity models. As a consequence the tool should be able to represent the different model structures and analysis processes.

Support for different scale levels: Not all maturity models work with ratio scales, but use other scales such as the ordinal scale. Therefore, the tool should be able to work with different scale levels.

Connectivity: The tool should provide an interface for external applications to access existing data sources, such as process repositories and process analysis tools. Thus, not all information must be manually entered into the tool. On the other side, also the process analysis tool can benefit from that connection by using the results of the maturity assessment to define focus areas for a deeper evaluation.

An analysis of the business process management tools described in Chapter 2 with regard to the identified requirements was not done in detail as none of these tools is generic or can be really configured by the user.

4. Development of the Prototype

4.1. Basic idea for realization

The following maturity models are selected examples for the design of the tool: CMMI, SPICE, PEMM, innoGov [12] and BPM Maturity Check. Our analysis of these models has shown that all tested models have in their specifications so-called objectives that an organization must achieve in order to attain a certain level of maturity. The models differ in the specific objectives and in the way how they evaluate the maturity level.

On the basis of this result, a questionnaire for each of the above maturity models can be developed which contains questions to the specific objectives of each model. Another result of this study is that developing a universal questionnaire valid for all models is not possible. The objectives of the individual models are strongly focused on the underlying concepts and evaluation guidelines. If a universal questionnaire was used, the results would be inconclusive.

The definition of the questionnaire includes that several answer options for each question are developed, representing different degrees to which the objective is achieved. These response options are structured according to a consistent pattern to enhance the readability.

The condition for the automatic maturity evaluation is that scores for all possible answer options are defined first. The score height represents the achieved level of the corresponding objective. The higher the achieved level of the objective, the higher is the given score. In the basic configuration of the prototype, one specific maturity level is represented in all answers by the same score. Thus, all objectives are equally weighted, but a question may be re-weighted without any further changes in the tool. The flexible computing system automatically determines the maximum score for a question and compares the points obtained with the maximum value to calculate the ratio.

To handle the different ways how to evaluate the maturity, an individual evaluation page is developed for each model with regard to the specifications given by the model. For maturity models for which the evaluation is not explicitly defined, plausible guidelines are developed. The ratio scale can be used directly with the calculated results. For all other scales, the transformation into the corresponding scale level is made by the specific result page.

4.2. Functional design

The necessary functions of the tool can be derived from the requirements of Chapter 3. During the requirements analysis, we identified the need for administrate functions, functions to create or adapt a
maturity model and functions for the usage of the tool to execute assessments. In Figure 2, the functions of the tool are presented in a function tree. It fulfills the functional requirements for the generic tool as described in the bottom part of Table 2.

### 4.3. User profiles

For the usage in a real environment, it is necessary to define different user profile with corresponding user rights. According to our view, 4 different profiles are needed (see Figure 3). First, the administrator has all rights: he can create new users and clients, access all models, and watch any reviews and evaluations. Second, the normal user has all rights for one specific model: he can edit this specific model and view any associated assessments and evaluations, but he cannot create new users and clients. Third, the restricted user can carry out assessments and look at shared assessments and evaluations, but he cannot create users or clients and cannot edit the model. Forth, the reading user has only read access on shared assessments and evaluations, but cannot do any rating and has no other rights.

### 4.4. Data design

Based on the requirements presented in Chapter 3, the data model of the tool has to support the following aspects:

- Management of clients and users,
- Management of different maturity models with their objectives, response options and results,
- Storage of the response of each user and association with the corresponding model,
- Extensibility to other models without any conceptual changes.

The data model is developed using the Entity Relationship Model (ERM) (see Figure 4).

### 4.5. Technical Concept

Three architectural options are available for the implementation of the tool:

- A monolithic solution using Java,
- a web-based solution using PHP and
- a web-based solution using Ajax.

In Figure 5, the three architectural variants are presented.

In the monolithic system using Java, the application is running directly on the user's computer and accesses a local database. The surface is implemented with a Java GUI. This solution does not need any access to the internet and provides high flexibility in the design of the user interface. But it requires the installation of a Java engine on the user’s computer.

In the web-based solution using PHP, the application is running on an Apache server and accesses a central database from there. The interface is implemented using HTML pages and is viewable in a web browser. This solution promises a high user acceptance as most users are familiar with HTML pages. The solution can be easily adapted at any time by changing the code on the server. The central data repository can be used for benchmarks. The disadvantage of this solution is the need for internet access and for server capacity.

In the web-based solution using Ajax, the comparison of PHP and Ajax shows that Ajax
Figure 4. Data model

Monolithic Solution using Java

Web-based Solution using PHP

Web-based Solution using Ajax

Presentation Layer

Application Layer

Database Layer

Figure 5. Representation of the three architectural options
supports an asynchronous communication linked with a shorter response time and a reduced workload for the server. But Ajax requires the installation of a special engine on the user’s computer, causing more effort during the implementation. In addition, Ajax works only with selected web-browsers.

Because of the advantages and disadvantages of the three architectural options and because of the general requirements such as speed and ease of implementation, the web-based solution using PHP is chosen for the prototype.

4.6. Implementation

Figure 6 shows the system architecture of the tool. The application runs on an Apache 2 server where PHP is installed. It is accessible via the internet from any client computer with a web-browser.

The user interface is provided by HTML pages. In Figure 7, you can see the result of a maturity assessment with the BPM Maturity Check.

A MySQL database is used as data storage.

4.7. Application

In this chapter, it is presented how the CMMI maturity model is applied by the tool. CMMI is implemented in accordance with the guidelines of the CMMI-DEV v1.2. CMMI-DEV provides a comprehensive and integrated maturity model for development and maintenance activities for products and services. As part of the prototype development, CMMI-DEV is not fully implemented, but only the selected parts “Requirements Management” and

<table>
<thead>
<tr>
<th>Table 3. Scheme of CMMI answer options</th>
</tr>
</thead>
<tbody>
<tr>
<td>Answer Option</td>
</tr>
<tr>
<td>---------------</td>
</tr>
<tr>
<td>No, there is no …</td>
</tr>
<tr>
<td>… has been started.</td>
</tr>
<tr>
<td>Yes, there is …</td>
</tr>
</tbody>
</table>
Table 4. Example for the implementation of one CMMI requirement

<table>
<thead>
<tr>
<th>Requirements</th>
<th>Implemented Issue</th>
<th>Answer Option and Points</th>
</tr>
</thead>
<tbody>
<tr>
<td>SP 1.1 Establish Measurement Objectives.</td>
<td>Was a common understanding developed with the requester about the importance of the requirements?</td>
<td>No, there is no development of an understanding with the requester about the importance of the request. This response option is evaluated with 0 points because there is no implementation of the requirement.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>The development of an understanding with the requester about the importance of the request has been started. This response option has a score of 3 points, since the implementation of the requirement is already in work.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Yes, there is a development of an understanding with the requester about the importance of the request. This response option is evaluated with 5 points because the requirement has been implemented.</td>
</tr>
</tbody>
</table>

“Project Planning” on Maturity Level 2 called “Managed” as well as “Requirements Development” and “Technical Solution” on Maturity Level 3 called “Defined” [2].

Based on the objectives and the maturity evaluation process of CMMI, a catalogue of questions with corresponding answer options and a specific evaluation page were developed. For each question, the CMMI questionnaire defines three answer options. These options are built according to the scheme represented in Table 3 and are measured by the indicated points. Table 4 shows an example for the implementation of one selected requirement.

All questions are assigned to a generic or to a specific objective. An objective is fulfilled if the questions that are associated with it obtain at least 85% of the maximum possible score [2]. For the maturity evaluation, the cumulative evaluation of CMMI has to be recognized [6]:

- To achieve the Maturity Level 2, the object must fulfill all specific and all generic objectives of the level 2.
- To achieve the Maturity Level 3, the object must fulfill all specific and all generic objectives of the level 3. In addition, the Maturity Level 2 must be also achieved.

Analogous to the application of CMMI, we successfully implemented also SPICE, PEMM, BPM Maturity Check, and innoGov in our prototype. These maturity models were selected for the prototype implementation because they are widely distributed in practical usage and because they represent different structural aspects and contents which we wanted to test (e.g. ratio scale of CMMI vs. ordinal scale of PEMM; difference of capabilities and maturities in CMMI vs. only capabilities in SPICE; more traditional models vs. model adapted to technical realization like BPM Maturity Check; assessment of business processes vs. focus on public administration in innoGov).

5. Evaluation

5.1. Example Usage Scenario

One maturity model implemented in the prototype is innoGov. Due to this fact, the tool can be used in our project to improve the 48-h-service promise in the German state administration. The target of that project is that “incoming requests or applications to public authorities shall be responded within 48 h by a confirmation of receipt or an official notification, at the latest.” [12] During a case study, we implemented the developed maturity model innoGov in one selected state administration. The implementation of innoGov could be supported by using the tool. With the help of the tool, it is possible to document the status at the beginning of the project and to identify the specific improvement potentials of each single department. Furthermore, it is possible to do a quick assessment at any time to document the improvement progresses and to check the points still open. That leads to better information flow to the involved employees in the state administration, to a higher engagement of the responsible persons and thus to a faster achievement of the project targets. This usage scenario proofs the usefulness of our prototype and the fulfillment of our intensions.

5.2. Design Science Research Approach

The development of the tool followed a Design Science Research approach. According to that approach, the boundaries of human and organization capabilities are tried to extent by offering new IT opportunities. To ensure the scientific value of the study, the Guidelines for Design Science Research by Hevner et al. [4] were followed.

Guideline 1 – Design as an Artifact: The artifact of the study is the generic tool for the application of
maturity models. Following the definitions of March and Smith regarding the classification of artifacts into constructs, models, methods and instantiations [8], we can specify the tool in more detail: The tool itself can be considered as a construct. Standard methods of software development (e.g. like function tree and data model) were used to design the tool. The prototype represents an instantiation of the tool.

**Guideline 2 – Problem Relevance:** Maturity models are often used for assessment of business processes because they identify vulnerabilities and potentials for optimization of processes [6, 9]. Their application is very complex and time consuming as, for example, the client’s employees have to be interviewed and the collected data has to be evaluated by experts to determine the maturity. The generic tool developed in this study supports organizations in determining their maturity levels by facilitating and automating the maturity evaluation.

**Guideline 3 – Design Evaluation:** According to Hevner, the usage of well-executed evaluation methods is a crucial component of the research process [4]. Therefore, the study included an analytical static analysis by examining the structures of the maturity models and reducing them to a common basis. The development of the prototype to demonstrate the application of several different maturity models as well as the usage of different scale types was executed as a controlled experiment. The usage scenario can be interpreted as a case study.

**Guideline 4 – Research Contributions:** The development of the generic tool represents a benefit for research because a general basis of maturity models was found and a way how to implement different maturity models in one generic tool was created. That basis can be used for further research in the area of maturity models.

**Guideline 5 – Research Rigor:** The basis for the development of the artifact was an extensive basic research on maturity models [3]. The development of the tool followed the methods of the classical software design process, e.g. well-proven methods like ERM were used for the documentation. After all aspects like requirements and design of the tool were defined and documented, the implementation of the prototype started.

**Guideline 6 – Design as a Search Process:** The study was designed as an open investigation. The exact ideas how to handle the concrete problems were developed only during the study. The aim of the work was to develop a pragmatically feasible solution. The development of the generic tool was iteratively done in the course of a search process which was marked by several phases of development and returns.

**Guideline 7 – Communication of Research:** The project is described in an internal report of our research organization. Also a presentation to interested colleagues and students was given. The next step is the presentation and discussion on relevant conferences.

As all guidelines of the Design Science Research were appropriately followed, we are convinced of the scientific value of the study.

5.3. Evaluation of the Solution

The aim of the study was that our tool should enable a simple, convenient and automated determination of the maturity level of an object or an activity. We defined two criteria for that aim: First, a user who has only basic knowledge of a certain maturity model should be able to quickly and easily determine the maturity level of an object or an activity. Second, the tool should automate processes and analyses.

The first criterion is realized by our basic idea that the user has only to answer questions by choosing the appropriate answer option. The complete maturity evaluation is done by the tool. The result of the maturity assessment is presented in an easily understandable graphical representation with adequate explanations.

The second criterion is realized especially by the automatic calculation of the result of the maturity assessment.

A prototype was implemented to prove the design of the tool and to allow practical tests of the application. Due to short resources capacities, the maturity models were not completely implemented. The limitation on selected parts (e.g. only one or two areas and/or maturity levels) offered the opportunity to test a wide range of maturity models. With regard to the detailed requirements derived from our aims and described in Chapter 3, the developed prototype fulfills all requirements with exception of the connectivity to other external applications. In the next iteration of the project which has been recently finished, we have examined the integration of a tool for maturity model assessments with external applications for business process management. During that phase, we could successfully develop a concept which type of data can be automatically extracted from a business process management tool as described in Chapter 2. In addition, we have extended the prototype of this study to extract data directly from ARIS platform. The results of that project will be published soon.
6. Conclusion and Outlook

Today, maturity models have taken an important role and its importance will increase in the next years. Therefore, tools that perform automated determination of a maturity level and that can also deal with several different maturity models are urgently needed.

The result of our study was that we could develop a prototype which can be used by users with only basic knowledge about maturity models, which is able to handle different maturity models and which supports an automated determination of the maturity level degree of an organization.

During the study, we discovered two important findings. In our view, the main finding is that the examined maturity models can be reduced to a common basis despite their complex structure and their different objectives. As all models are dealing with objectives that must be achieved in a specific area to achieve a certain maturity level, lists of questions and answer options can be created representing the objectives and the degree of their fulfillment. Based on that finding, we could construct a database schema which represents all maturity models.

The second finding is that different evaluation scales can be presented with only minor adjustments. The tool can be used for both ratio scales (e.g. CMMI) and ordinal scales (e.g. PEMM). The ratio scale can be calculated directly from the maximum possible score. The ordinal scale uses the scoring to set the selected answers in the right order.

The research in the field of generic tools for maturity assessments is still at its early stages. Generic tools have great potential and could become an essential support for maturity assessments in future. For this purpose, it is necessary to further develop the tools in order to work more automated with access to databases, simulation tools, or ERP (Enterprise Resource Planning) systems. The stronger automation will lead to a higher standardization and less manual work at the same time. The next generation of maturity models could be intelligent maturity models. These intelligent maturity models are fully integrated into a business process management system and can extract and analyze the stored business processes. In addition, intelligent maturity models can generate enhanced optimization suggestions by using expert systems. In return, the BPM system can use the results of the intelligent maturity model to highlight and improve business processes that have a need for optimization.

As soon as the development of generic tools for maturity analysis is further advanced, these tools could emerge as a central element in maturity assessment.

7. References


