[vem:xi:] - A methodology for process based requirements engineering

Philipp Liegl, Rainer Schuster,
Marco Zapletal, Christian Huemer, Hannes Werthner
Vienna University of Technology
Institute for Software Technology and Interactive Systems
Vienna, Austria
{*firstname.lastname*}*@tuwien.ac.at*

Michael Aigner, Martin Bernauer, Bjoern Klinger,
Michaela Mayr, Ramin Mizani, Martin Windisch
Austrian Mobile Network Operator
Vienna, Austria
vemxi@gmx.at

Abstract

Service-oriented architectures (SOA) aim at the alignment of business and IT by having a clear business process-centric focus. In order to reach that goal, real-world business processes are captured by business process models. These models serve as the basis for the declarative configuration of a SOA using appropriate deployment artifacts - i.e., XML-based process languages. Consequently, requirements engineering for SOAs must focus on business processes and on their integration into systems using interoperable services, which is not the case for most conventional requirements engineering approaches. In this paper we present a requirements engineering approach specifically designed for the engineering of SOAs. Requirements are captured using a unified process, based on phases and iterations eventually leading to a formalized and unambiguous requirements specification. The final requirements specification can be used in succeeding development phases - i.e. for the model-driven generation of deployment artifacts for SOAs. The presented solution is called [vem:xi:] and is successfully used in the IT department of a Mobile Network Operator (MNO) in Austria. In the highly volatile world of mobile communication the presented approach enables faster application development and faster integration of solutions, thus leading to a competitive advantage over other market participants.

1. Motivation

The introduction of a new software solution in an enterprise contains a set of reoccurring challenges. In particular the requirements engineering process is error-prone and contains a set of potential risks [1]. Before a solution is introduced the different stakeholders such as business domain experts, business analysts, user interface designers, developers, the management etc. must have a common understanding of the solution to-be. However, most of the stakeholders define their requirements using different concepts, languages, and tools. Thus, the final requirement specifications are often redundant and incompatible.

Requirements engineering is a dynamic process which does not consist of self contained phases and steps but is rather executed in an iterative and repeating manner. Since most requirements engineering processes in companies are rather rigid and inflexible, requirement changes can only be partially reflected or are not considered at all. Furthermore, when a new solution is introduced in an existing system landscape there are a multitude of interdependencies to existing processes, systems, and other solutions. Most of the known requirements engineering approaches do only consider the new solution architecture's requirements. Little to nothing is specified in regard to interdependencies to existing solutions and systems.

Finally the advent of the service orientation paradigm has brought additional challenges to the requirements engineering domain. Service oriented architectures (SOA) are expected to deliver a flexible alignment between business and IT. The goal of business/IT alignment is achieved by SOA services that realize business processes. A business process is essentially a semi-formalization of business needs and requirements. Although these benefits are well known, a specialized requirements engineering process for the proper design of a SOA is still missing. Existing requirements engineering approaches do not completely bridge the gap between business and IT in order to reach a proper alignment.

In the paper at hand we provide a new technical solution aiming to overcome the limitations mentioned above. We introduce our business process based requirements engineering approach called [vem:xi:]. In our approach we use the Business Process Modeling Notation (BPMN) which currently as being state-of-the-art in both, industry and academia. BPMN provides a standardized mapping to the Business Process Execution Language (BPEL). BPEL is a declarative process specification language which is used to configure execution engines accordingly. Consequently our approach provides the basis for the generation of SOA artifacts out of the artifacts gathered in the requirements engineering phase. Current process based requirement engineering approaches often do not meet these criteria.

The remainder of this paper is structured as follows: section 2 gives an overview of related work in the field of process based requirements engineering. Section 3 introduces the basic concepts of [vem:xi:] and section 4 introduces the accompanying example. In section 5 the different phases of [vem:xi:] and their artifacts are elaborated in detail. Finally section 6 concludes the paper and gives an outlook to future research.

2. Related Work

An overview about the combination of different requirements engineering approaches and their dynamic selection depending on the project context has been presented in [2]. The survey summarizes techniques having a strong focus on the stakeholder's involvement in the requirements engineering process. Another analysis of collaborative requirements approaches as proposed by [vem:xi:] is also given in [3] and [4]. The first one proposes an integrated model in order to create a graphical representation of an analysis model in an early design deliberation phase. The latter describes a method on how to filter out the necessary information from collaborative workshops with stakeholders in order to elicit the requirements. In contrary to [vem:xi:], both approaches cover only the requirements engineering techniques tailored to the use in an early development stage, in order to get a first sketch of the IT system to be designed. Gruenbacher [5] proposed a methodology for collaborative requirements engineering. The approach is based on a UML meta model which is used to capture the different viewpoints of stakeholders. Furthermore the author used a meta model to investigate different tools in regard to their requirements engineering capabilities. However, since [vem:xi:] uses UML concepts as well, there are some overlaps between these approaches. A short article questioning the role of a requirements engineer is given by Paech [6]. Potential mismatches in the requirements definitions for ERP systems are identified by Daneva et al. [7]. A thorough examination of the gap between classical requirements engineering approaches and process based requirements engineering has been made by Arao et al. [8]. In their paper the authors provide a new requirements information model and requirements engineering process. However, the authors are missing a formalized process model allowing a model driven approach towards software artifact generation. A requirements engineering approach which focuses on the visualization of requirements has been presented by Pichler et al. [9]. Thereby the authors introduce a business process based requirements engineering approach and evaluate the tool integration of their approach. In contrast to this approach, [vem:xi:] has a formalized process based on phases and iterations. Furthermore, [vem:xi:] delivers a final requirements specification which is tailored for a model-driven generation of deployment artifacts for SOAs.

3. The [vem:xi:] methodology at a glance

In order to overcome the limitations mentioned in the introduction of the paper at hand the [vem:xi:] methodology has been introduced at an Austrian mobile network operator to which we in the following refer to as MNO. As shown in figure 1 the methodology consists of six distinctive phases: value proposition, environmental analysis, macro planning, micro planning, GUI design, and validation/simulation. Each phase has well defined objectives and delivers a set of artifacts and documents. The name [vem:xi:] is an acronym of the first letters of each of the six phases: VEMMGSi - phonetic transcription [vem:xi:]. The different phases of

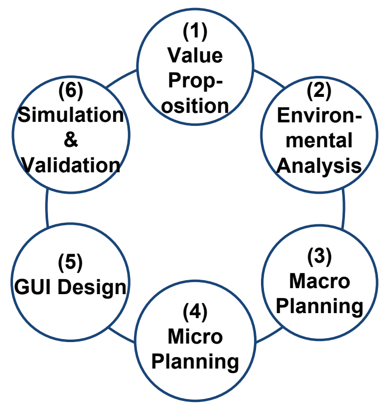


Figure 1. The [vem:xi:] methodology

[vem:xi:] are not self contained but have strong interdependencies. Each phase has a distinct goal and its artifacts serve as input for the following phases: *Value proposition* - Specify the value and the purpose of the solution which should be introduced. *Environmental analysis* - Identify the affected entities, side affected entities, affected projects and systems as well as organizational units involved in the solution. *Macro planning* - Use the input of the two previous phases to construct a first coarse grained process model. *Micro planning* - The coarse grained process model is gradually refined to the final process model. *GUI design* - For each user interaction a set of GUI mock-ups is assigned to the relevant process activities. *Simulation & Validation* - The designed model is verified using process simulation concepts.

As shown in figure 2 the six phases of [vem:xi:] are split up into four steps, similar to the rational unified process (RUP) [10]. Each step contains a set of iterations which are executed over and over again until the final artifacts per phase are finished. At any point within the [vem:xi:] lifecycle a going-back to a previous phase or step is possible. Depending on the different steps, each phase in [vem:xi:] has either more or less significance throughout the overall construction of a [vem:xi:] model. E.g. value proposition is important in the first two steps inception and elaboration but is of minor importance in the last two steps construction and transition. Each of the different steps serves its own purpose

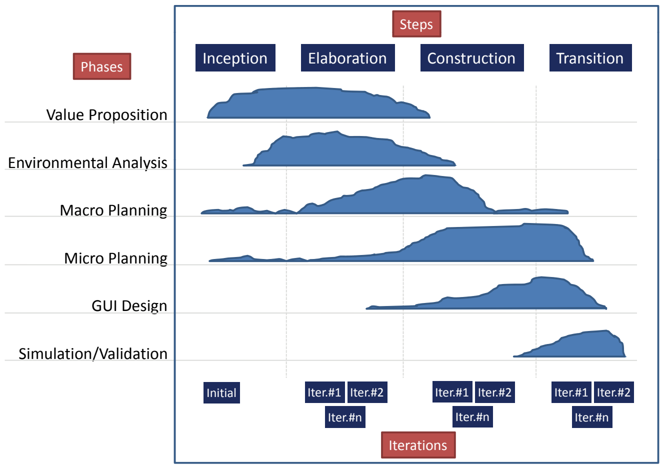


Figure 2. Iterations within [vem:xi:] phases

and a set of well defined actions is taken per phase:

Inception. In this step the value proposition is created together between business analysts, business domain experts, and the management. The first artifacts for the environmental analysis are constructed.

Elaboration. Following the inception, the elaboration step further refines the artifacts created in the previous step. The value proposition and the environmental analysis are finished in this phase and the first coarse grained models are constructed. In case any inconsistencies in regard to the proposed values or environmental conditions are found, artifacts are adapted accordingly.

Construction. In the construction step the macro planning artifacts are further refined and specialized resulting in a fine-grained model. The fine grained model is further equipped with GUI mock-ups. Similar to the elaboration phase the modeler has to ensure that the created fine-grained model is in accordance with the artifacts specified in the value proposition and environmental analysis phases. If necessary step 1-3 are re-iterated.

Transition. During the transition phase the created artifacts are validated against the real world scenario using simulation and validation techniques. If any inconsistencies are found the business analyst initiates another [vem:xi:] iteration in order to adapt the created model.

The different phases of [vem:xi:] are interweaved to a different extend. The phases micro planning, macro planning, GUI design, and simulation/validation are strongly connected whereas value proposition and environmental analysis are related to a weaker extend to the other phases.

It is important to notice, that changes in the overall process design late in the [vem:xi:] project are more cost intensive than in early phases of the project. This term is known as *late-design-breakage*. Especially in the early phase of a project, the requirements for a new software solution often and regularly change. The iterative approach provided by the [vem:xi:] methodology aims at capturing these changes at any time and at reducing the overall costs induced by the changes (late-design breakage costs).

As shown in figure 3, the business process management domain is split up into four distinctive parts: process modeling, process development, process execution, and monitoring. The [vem:xi:] methodology covers the first part of the framework - process modeling. [vem:xi:] allows the description of end-to-end flow activities across organizational units. The [vem:xi:] model reflects the business view and is an important tool for the business owner, in order to discuss and communicate the needs of a new (or modified) process. The final output of the process modeling phase is a

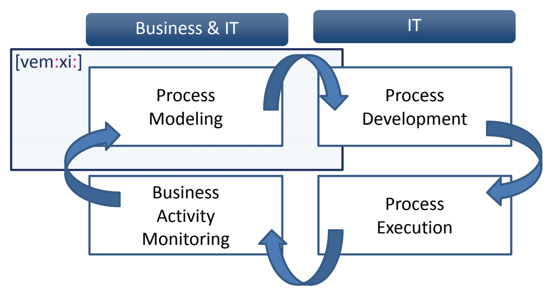


Figure 3. [vem:xi:] and the business process lifecycle

validated and fine-grained process model, uniquely defining the requirements of the solution to-be. Such a process model is further refined during the process development phase where e.g. concrete service definitions are assigned to the different process activities. The final process model of the process development phase may serve as input for execution engines during the process execution phase. Whether the overall goals as specified in the value proposition phase of a [vem:xi:] process are met or not is determined during the business activity monitoring phase.

In order to ensure that the difference between the actual results of the process execution and the defined goals in the value proposition phase is as low as possible, the early validation of the [vem:xi:] model is of utmost importance. A first high-level simulation and validation as part of phase six of [vem:xi:] helps to apply first optimizations early in the design process.

4. Accompanying example contract take-over

The [vem:xi:] methodology itself is tool independent and an implementation can be made with any business process modeling tool of choice. Depending on the features provided by a specific tool some phases of [vem:xi:] might not be able to be realized. For the accompanying example of this paper the business process modeling tool Enterprise Architect [1] has been chosen since it is widely available at low cost.

In the following, the concepts of [vem:xi:] are introduced using a real world example called ”contract take-over”. In case an existing customer of the mobile network operator (MNO) wants to transfer his or her contract to another customer (either an existing customer of the MNO or a new customer), an MNO point of sale is contacted. We refer to the customer to whom the contract gets transferred to as `consignee` and to the customer who transfers his contract as `consignor`.

The consignor request a PIN from the MNO which he communicates the consignee. Using the PIN from the consignor the consignee can contact a MNO point of sale and requests the contract take-over. Until now the contract take-over procedure within the MNO included several manual steps and media breakages (e.g. sending a fax to the MNO head office and manually transferring the data into the backend system). The manual notion of the process proved to be error prone and slow, thus lowering customer satisfaction. In order to overcome these limitations a new, automated approach for the contract take-over has been introduced at the MNO. To ensure that all requirements are met [vem:xi:] has been chosen as the methodology of choice for the introduction of the new IT solution. In the following sections the contract take-over example is used to outline the different artifacts of every [vem:xi:] phase.

Responsibilities of the different roles involved in the [vem:xi:] approach are indicated using the RACI matrix. The RACI approach splits responsibilities into four different responsibility types. The different types are then assigned to the roles within the [vem:xi:] approach: R - responsible for producing deliverables. A - accountable for quality and timeliness of deliverables and ensuring that key people are consulted. C - must be consulted in the production of the deliverable. I - inform i.e. receive a copy of the deliverable.

5. [vem:xi:] by example

The following sections examine the six different phases of [vem:xi:] in detail and give a detailed overview of the used objectives, scope, methods, and tools. For every phase an example from the accompanying example *contract take-over* is shown.

5.1. Value Proposition

The main objective of the value proposition phase is the business justification. In order to justify the introduction of a new solution, its purpose and benefit must be specified - in a business sense. If an existing system has to be replaced, the costs for the replacement must be captured in a structured way. The replacement details in terms of the affected entities and organizational units are further specified in the next phase of [vem:xi:].

The scope of the value proposition phase covers the definition of the value delivered by the IT-solution. Furthermore the goals and non-goals of a solution are elaborated and documented. All goals defined in this phase must be in accordance to the overall IT goals of the company and in alignment to the overall IT strategy.

The value proposition phase uses a set of different methods to capture the necessary information. In order to get the required input for the value proposition phase, brainstorming sessions are held by the business analysts to get a first overview of the values and benefits of the introduction of the new IT-solution. However, the actual value propositions are not only decided by the business analyst but by the different departments and units within the enterprise. The first preliminary results of the business analysts’ brain storming sessions are used to generate questionnaires and interview road-maps. Interviews and opinion-polls using the questionnaires are held together with stakeholders from the different departments. Eventually this leads to a common understanding and agreement of the values the new solution is supposed to deliver for the participating stakeholders. Since all stakeholders are included in the requirements process at this early phase misunderstandings and wrong expectations are prevented. In case the new solution has direct customer interaction, the specific customer requirements are reflected accordingly. These requirements and expectations can for instance be collected by online surveys or telephone surveys.

For capturing the specific value proposition requirements the business analyst has a set of tools at his disposal including mind mapping tools, word processing tools, and project management tools.

For the [vem:xi:] contract take-over showcase the artifacts of the value proposition phase have been elaborated using a mind mapping tool. Figure 4 shows a cut-out of the value proposition mind map. The business analyst has split up

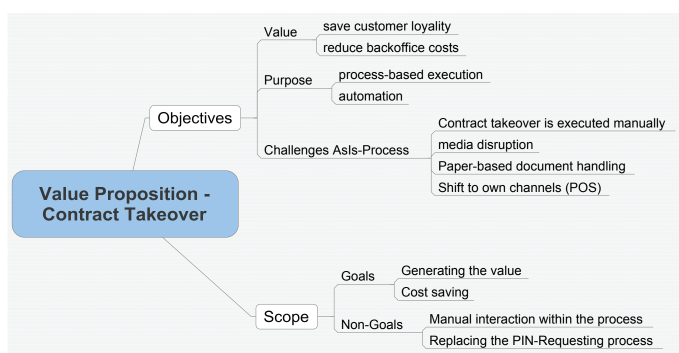


Figure 4. Cut-out of the value proposition mind map

the objectives of the new solution into three subcategories: value, purpose, and challenges for the existing processes.

1. http://www.sparxsytems.com.au

Target entities	Affected entities	Side-affected entities
Products	Products	Products
Customer segments	Systems	Systems
Channels	Processes	Processes
	Organizational Units	Organizational Units
	Projects	Projects

Table 1. Classification scheme for environmental analysis phase

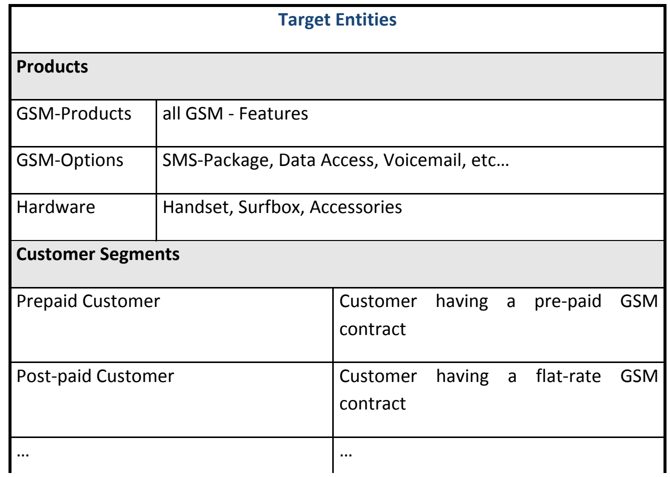
The scope has been divided in goals and non-goals. According to the RACI specification the involved roles in this phase are: Business (R), Business Analyst (R/C) and Management (A).

After the expected values of the new solution have been captured and an agreement of the new solution in a business sense has been made, the business analyst proceeds with the environmental analysis phase.

5.2. Environmental Analysis

The main goal of the environmental analysis is to define all entities which may be affected by the introduction of the new solution; in other words, how the new solution fits into the existing process and system landscape. Within [vem:xi:] a distinction is made between three different types of entities namely target entities, affected entities, and side-affected entities. Target entities are involved in the new solution, no matter if they already exist or need to be introduced. These entities are the main focus and the purpose for the introduction of the new solution. If the introduction of a new IT solution has an influence on entities which are part of other solutions, these entities are called affected entities. In contrast to a target entity an affected entity must already exist. Entities which are part of the new solution and other solutions but remain unchanged by the introduction of the new solution are called side-affected entities. Depending on the actual context where [vem:xi:] is applied, a further sub-classification of the different entities can be made if necessary.

The scope of the environmental analysis phase includes the collection of all necessary entities and their classification. Although [vem:xi:] does not mandate to use a specific classification the hierarchy as outlined in table 1 is generally recommended. The process classification results in a set of categorized worksheets as shown by the cut-out in figure 5. In a second step the business analyst captures the identified entities using a business process modeling tool. The representation of the different entities within the modeling tool is important, since in later steps the entities will be assigned to process activities. Thereby responsibilities of organizational units and interdependencies between the new solution and the other business environment can be shown. It is important to notice, that all decisions made in this phase have to be



Target Entities	
Products	
GSM-Products	all GSM - Features
GSM-Options	SMS-Package, Data Access, Voicemail, etc…
Hardware	Handset, Surfbox, Accessories
Customer Segments	
Prepaid Customer	Customer having a pre-paid GSM contract
Post-paid Customer	Customer having a flat-rate GSM contract
…	…

Figure 5. Classification example

in accordance with the goals of the overall process strategy. This means, that no existing systems, solutions or business cases must be altered in such as way, that the overall process strategy is contradicted.

The methods and tools used in this phase include interviews, questionnaires and workshops. The collected information is held in worksheets and transferred into a formalized model representation. The cut-out in figure 6 gives an overview on how UML packages can be used to structure and classify the identified entities. According to

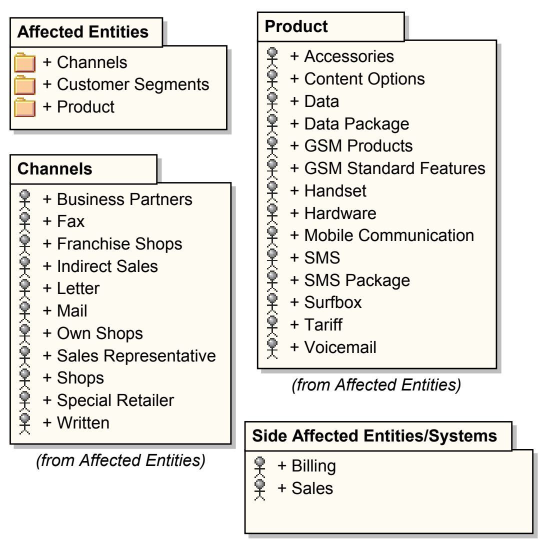


Figure 6. Package structure cut-out

the RACI specification the involved roles in this phase are Business (R/C), Business Analyst (R), Management (A), IT/Architecture (C/I), and Solution Designer (C).

After the business analyst has successfully completed the value proposition phase and has analyzed the environment of the new solution he starts to create a first coarse-grained model.

5.3. Macro Planning

During the macro planning phase the concepts of the first two phases are aggregated to a formalized model. Thus the main objective of the macro planning phase is the construction of a coarse grained model. The required processes must be identified and an overview of the most important steps and used resources is created. Furthermore the first model sketch can be used for an early management review.

The scope of the macro planning phase includes the creation of a process list. Process lists represent an ordered set of process activities and sub-activities involved in the new solution. Activities and sub-activities are structured in must-have, should-have, and nice-to-have schemes. Relevant existing processes within the company and new to-be created processes are identified. The delta between existing and required processes has to be analyzed in order to give a first overview of the implementation complexity. Again it is important to notice, that all decisions made in this phase have to be in accordance with the goals specified in the first phase of [vem:xi:].

Figure 7 shows a cut-out from the example macro model process list. Please note, that the categories on the right hand side follow a color code, which is also reflected in the macro planning model. The second column of the process list denotes the name of the (sub)-process activity. Each activity has a certain goal which is described and outlined in the third column called description and goals. An activity is finished after the goal of the activity has been fulfilled. The macro planning phase uses a set of

ID	Process Name	Description and Goals	must have	should have	nice to have
P1	LogIn	The user authenticates himself using a valid username/password combination. After a successful login the user can use the Sales Frontend functions			
[...]					

Figure 7. Process list of the macro planning phase

different methods and tools to create the necessary artifacts. The process list is created using a spread sheet tool of choice e.g. Microsoft Excel. In the next step a business process modeling tool is used to create the actual process model and worksheet information artifacts. A worksheet defines the exact structure, how process meta-information is aligned, classified, and stored. For the purpose of the [vem:xi:] methodology an XML schema is used in order to define the conceptual model of a worksheet. Business process management suites from tool vendors such as Oracle or IBM allow for the integration of XML schema artifacts into process models. The tool processes the XML schema artifacts and automatically generates the necessary forms for the business analyst where requirements information can be entered. Eventually the requirements information is stored in XML instance documents. Since all documents have been derived from the same XML schema, interoperability between the different document instances is guaranteed. Finally the XML documents are linked to the relevant activities and sub-activities in the business process model. Therefore necessary meta-information is directly connected to the relevant artifacts in the model.

The worksheet XML schema guarantees that all worksheet instances follow the same structure and can easily be attached to process artifacts. Furthermore the XML representation of requirements information allows for an automatic processing using appropriate tools.

The currently used tool (Enterprise Architect) does unfortunately not support the automatic form generation based on XML schema. However, Enterprise Architect allows to attach XML instance documents to activities and sub-activities in a business process. Thus a different tool had to be chosen in order to capture the text based requirements based on the same XML schema. We suggest to use Microsoft Word 2007 as the tool of choice for capturing worksheet data. Figure 8 shows the integration of XML schema into Microsoft Word 2007. First an XML schema

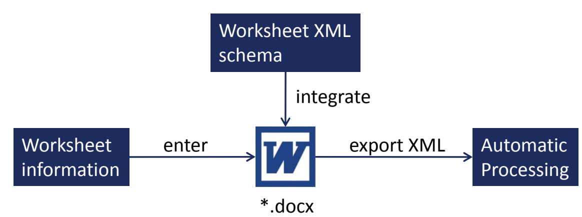


Figure 8. Integration of XML schema data into MS Word 2007

is associated with a Microsoft Word document. Due to the associated XML schema the business analyst is presented a predefined worksheet structure where only designated fields can be filled out. The entered worksheet information can either be used as regular Microsoft Word document (e.g. for communication purposes between business analysts) or exported as XML instance. In the following the exported XML instance is copied and attached to the business process model thus enabling automatic processing. The inclusion of XML schema artifacts into Microsoft Word is currently supported by version 2007 of Microsoft Word which supports the Office Open XML standard [11]. The major advantage of this approach is the common storage of free-form text requirements information, captured by worksheets, within business process models. Inconsistencies between requirements artifacts can be avoided and automatic processing of requirements information is enhanced.

After the business analyst has finished the worksheets and the process list, the macro model is created. In the following example the Business Process Modeling Notation (BPMN) is used in order to depict the macro model. Figure 9 shows a cut-out from the macro model of the contract-take over example. Within the macro planning model the different

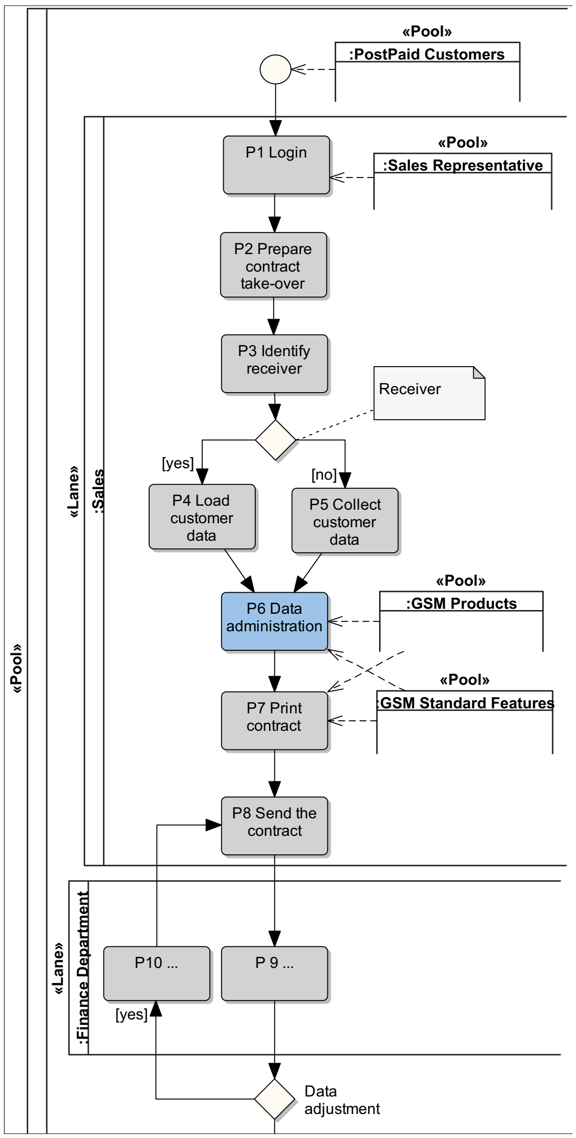


Figure 9. Macro model cut-out

process activities are aggregated in so called lanes. A lane indicates an organizational unit which is responsible for an activity. Lanes can be nested in order to depict organizational dependencies e.g. in figure 9 the organizational units *sales* and *finance department* are involved in the process. Target entities, affected entities, and side-affected entities which are involved in the process are modeled using pools. Pools are directly connected to activities where necessary e.g. *post paid customers* are invoking the process as shown on top of figure 9. It is important to notice, that the macro planning model's purpose is to give a brief overview about the solution to-be. No activities are refined in detail nor are any pre-or post-conditions defined or GUI elements assigned in this early stage. According to the RACI specification the involved roles in this phase are Business (R/C), Business Analyst (R) IT (C), and Management (A).

In the next step the business modeler refines the macro-model and adds additional, finer-grained requirements information to the model.

5.4. Micro Planning

The main goal of the micro planning phase is the further refinement of the artifacts created during the macro planning phase until the final business process model is finished. Furthermore the requirements documentation using the different worksheets and the process list is completed in this phase. Thereby the micro planning phase uses the same methods and tools as the macro planning phase - only the level of detail is finer-grained.

The first task for the business analyst is the refinement of the different worksheets. Since most activities from the macro planning model are split up into sub-activities, each sub-activity must have its own worksheet. In the next step the business analyst refines the process list from the macro planning phase. Thereby the different process activities are further elaborated using the concept of sub-activities. As shown in the process list in figure 10 the activity *login* from the macro-planning phase is split up into two sub-activities. The same principles as applied to the macro

ID	Process Name	Description and Goals	a/m	must have	should have	nice to have
P1	LogIn	The user authenticates himself using a valid username/password combination. After a successful login the user can use the Sales Frontend functions.	m			
P1.1.	Load Sales Frontend Login page	The necessary login page is loaded in the Sales Frontend.	a			
P1.2.	Logon procedure	The Sales Frontend checks the user credentials against the user database. If the credentials are valid the user is redirected to the Sales Frontend pages. Otherwise an error message is shown.	a			
[...]						

Figure 10. Micro model process list cut-out

process list are also applied to the micro process list. The second column contains the name of the (sub)-activity which is performed and the third column contains the concrete description and goal of the activity. Following the update of the process list the process model is updated accordingly. As shown in figure 11 the process model from the macro planning phase is extended and additional information is added to the model. The refined processes activities in the micro model are classified into manual and automated activities. An automated activity is executed by an IT system without any human interaction. A manual activity requires

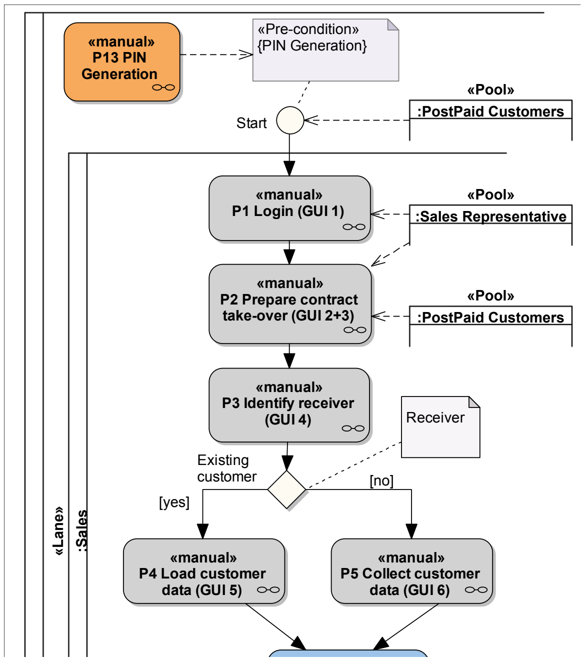


Figure 11. Micro model cut-out

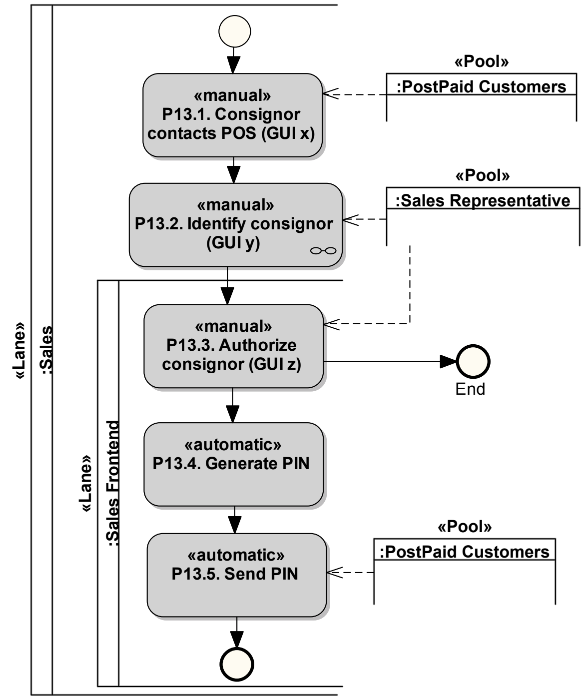


Figure 12. Sub-process PIN generation

human interaction. As outlined in figure 11 the concept of stereotypes is used in order to determine whether an activity is manual or automatic. All process steps shown in the cut-out require manual interaction. Again the color codes of the different activities help to differentiate between ”must-have”, ”should-have”, and ”nice-to-have” processes. Color codes are taken from the process list and are reflected accordingly in the process model.

Pre-conditions for activities or whole processes are indicated using the concept of constraints. As shown on top of figure 11 the pre-condition for the whole contract take-over process is a generated PIN. Thereby the process *P13 PIN generation* must be executed in order to allow the contract take-over process to start.

An organizational unit involved in the process is denoted using the concept of lanes. E.g. process steps P1 to P5 are executed in the context of the *sales department*. Affected entities, target entities, and side affected entities involved in the process are denoted using the concept of pools. E.g. process step P1 is executed by a *sales representative* and process step P2 additionally involves the *post-paid customer*. All organizational units and affected entities have been defined during phase 2 - environmental analysis.

An essential functionality of the micro-planning phase are composite activities. A composite activity is refined using another BPMN activity diagram. As an example figure 12 shows the activity diagram which refines the composite activity *P13 PIN generation* as shown on top of figure 11. Using this concept the business analyst can easily refine an existing macro model to the desired extend in order to meet the requirements of a micro model. Again the concept of lanes and pools is used for the different sub-activities. According to the RACI specification the involved roles in this phase are the same as in the micro planning phase: Business (R/C), Business Analyst (R) IT (C), and Management (A). The final micro-model is assigned with the necessary GUI mock-ups which are defined in the following phase.

5.5. GUI Design

The GUI design phase concerns the definition of GUI mock-ups for relevant process steps with human interaction. This means, that there is no technical implementation of any user interfaces required in this phase. The main objective of the GUI design phase is to demonstrate the look-and-feel of the solution. GUI mock-ups can be designed using any kind of graphical editor (e.g. Microsoft Visio). The output of this phase is a set of static and general user interfaces and a storyboard for a GUI sequence. The GUI mockups also play a major role in the micro planning phase. Each GUI must be assigned to a certain process activity in the micro model.

The design of the conceptual user interfaces is based on the collected information of the previous phases. According to the different process activities and sub-activities defined in the micro-planning phase, the GUI designer creates mock-ups for the different user interfaces. Naturally only manual activities which required user-interaction are assigned with GUI mock-ups. The mock-up design must consider state of the art principles e.g. accessibility standards, minimum font-size etc. Finally the GUI designer assembles the different GUI mock-ups to a final GUI storyboard as shown in figure 13.

The GUI storyboard for the [vem:xi:] example showcase was designed using Microsoft Visio 2007. Figure 13 depicts the flow of GUI, whereas the sequential flow of the different user interfaces is defined by the guarded transitions (e.g. [Login successful/failed]). This means, that the user is only able to get from GUI 1 to GUI 2, if he enters the correct login data. Each GUI symbol in the storyboard refers to a specific and accurately described GUI mock-up. Due to space limitations the actual GUI mock-ups are not shown. Eventually the different GUI artifacts are attached to the

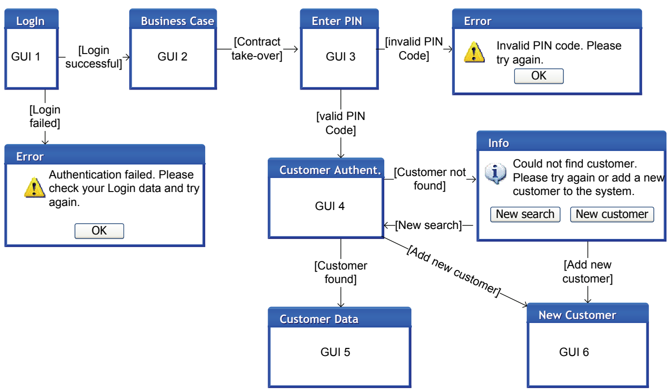


Figure 13. Contract take-over storyboard

micro-model activities and sub-activities. As shown in figure 11 every manual activity has the relevant GUI artifact assigned. E.g. the manual activity *P1 Login* requires the GUI element *GUI 1*.

According to the RACI specification the involved roles in this phase are Business (R/C), Business Analyst (I), IT/GUI Engineer (R), Process Owner (A), and Solution Designer (C/I).

After having defined the different user interfaces the process modeling phase is finished and the business analysts starts the final phase of the [vem:xi:] process - simulation and validation of the model.

5.6. Simulation/Validation

The final step in the [vem:xi:] approach is the validation and simulation of the constructed business process model. Business process simulation is based on so called discrete event simulation. Figure 14 gives an overview about the basic principles of a discrete event simulation. The simulation component interprets the business process model and simulates transitions of activity instances such as active or

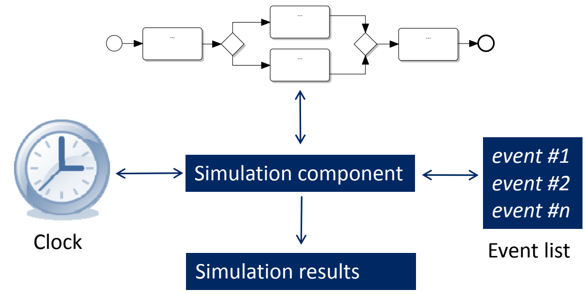


Figure 14. Discrete Event Simulation

terminated. It thereby removes an event from the event list, processes the event and causes changes to the model. Each event has a time stamp attached to it and the processing of events can generate new events which are automatically inserted into the event list. It is important to notice, that the clock advances from one discrete time to the next (in contrary to real-time simulation).

The main objective of the simulation phase is the evaluation of the designed model from the micro planning phase. It has to be ensured, that it is consistent with the results from the value proposition and environmental analysis phase. In regard to simulation the objective of this phase is the evaluation of the constructed model under different scenarios. A simulation of the process on the model level helps to apply first optimizations early in the design process. Furthermore possible deadlocks or synchronization errors can be identified and possible deviations from real world processes become apparent. Moreover first predictions can be made e.g. what is the behavior of the model if more resources are available, the process flow has changed or any given cost of an activity (time, money, resource) changes.

Before a model validation can be initialized, the process model must be complete and ready for simulation. The specific goals for the validation must be specified and be defined in the model. The process model itself must be stable enough for a validation. That means the necessary data annotations for the different resources and activities involved must be made e.g. time per activity, cost per resource etc. In general the appropriate simulation data must be available within the company. An example simulation run would at least include the following steps:

1) The different goals of the simulation must be defined precisely and it must be decided whether the simulation is feasible in terms of costs, available test data etc.
2) The existing data within the company must be analyzed and it must be decided whether the test data is appropriate for setting the simulation data. If the business process model is very complex, the simulation run should be split up into appropriate sub-processes.
3) The necessary data must be collected and aggregated.

4) The aggregate data must be incorporated into the process model.
5) Several simulation runs with different parameters should be executed in order to allow for a broader interpretation of the results.
6) The different strengths and weaknesses of the model can be identified using the simulation results.
7) If the overall process goals and the simulation results differ significantly, the appropriate changes have to be made to the model and the simulation must re-run.

In case the simulation results do not comply with the goals specified in the value proposition phase, the business analyst has to step back in the [vem:xi:] methodology and must apply changes to the macro and micro model respectively.

The simulation phase of [vem:xi:] depends on the capabilities of the tools used. In order to allow for a validation of the generated business process artifacts the business process modeling tool must support validation features. For the paper at hand Enterprise Architect has been used as the modeling tool of choice to create a [vem:xi:] compliant model. However, Enterprise Architect does not support the validation of a constructed model and therefore no concrete example for a model validation is given here. Tools supporting model validation of business process models include Oracle Business Process Management Tools, IBM WebSphere Business Modeler etc. According to the RACI specification the involved roles in this phase are Business (R), Business Analyst (R/C), IT (C), Management (A), and Solution Designer (C).

With the final validation and simulation of a [vem:xi:] model the process modeling phase as shown in figure 3 at the beginning of this paper is finished. The final process model is used for the process development phase where e.g. concrete service definitions are attached to the business process model, in order to use it as input for the process execution phase.

6. Conclusion

In this paper we have introduced our new technical requirements engineering solution [vem:xi:]. The presented, process-based approach helps to overcome a set of limitations of classical requirements engineering approaches. We have introduced the six phases of [vem:xi:] and have shown the application of the different phases using an example from the mobile communication domain. The [vem:xi:] approach has been evaluated during a successful application in an Austrian Mobile Network Operator's IT department. Through the process based requirements engineering the Mobile Network Operator is able to faster deploy new solutions, thus avoiding costs and increasing customer satisfaction.

As outlined in the paper at hand [vem:xi:] concentrates on the process modeling phases of the business process lifecycle. Future work will concentrate on the extension of [vem:xi:] in order to allow concrete service bindings to be added to the different activities and sub-activities - an activity typically performed in the process construction phase (cf. figure 3). This approach will require the introduction of a new layer after the micro-modeling phase and an extension of the simulation/validation phase. If [vem:xi:] can be successfully used in the process construction phase as well, an easier integration of the developed requirements artifacts into the process execution phase would be possible.

References

[1] L. B, K. Wiegers, and C. Ebert, "The top risk of requirements engineering," *Software, IEEE*, vol. 18, no. 6, pp. 62–63, 2001.

[2] D. Mishra, A. Mishra, and A. Yazici, "Successful requirement elicitation by combining requirement engineering techniques," in *Proceedings of Applications of the First International Conference on Digital Information and Web Technologies, (ICADIWT 2008), August 4-6, Ostrava, Czech Republic*, 2008, pp. 258–263.

[3] D. Ang, L.-H. Lim, and H.-C. Chan, "Collaborative requirements engineering: an overview and a proposed integrated model," in *Proceedings of the Thirty-First Hawaii International Conference on System Sciences, January 6-9, Hawaii, USA*, vol. 5, 1998, pp. 355–364.

[4] E. Gottesdiener, "Requirements by collaboration: getting it right the first time," *Software, IEEE*, vol. 20, no. 2, pp. 52–55, 2003.

[5] P. Gruenbacher, "Integrating groupware and case capabilities for improving stakeholder involvement in requirements engineering," in *Proceedings of the 26th Euromicro Conference, September 5-7, Maastricht, Netherlands*, vol. 2, 2000, pp. 232–239.

[6] B. Paech, "What is a requirements engineer?" *Software, IEEE*, vol. 25, no. 4, pp. 16–17, July-Aug. 2008.

[7] M. Daneva and W. R, "Requirements engineering for cross-organizational erp implementation undocumented assumptions and potential mismatches," in *Proceedings of the 13th IEEE International Conference on Requirements Engineering (RE2005), August 29 - September 2, Paris, France*, Aug.-2 Sept. 2005, pp. 63–72.

[8] T. Arao, E. Goto, and T. Nagata, "Business Process Oriented Requirements Engineering Process," in *Proceedings of the 13th IEEE International Conference on Requirements Engineering (RE 2005), August 29 - September 2, Paris, France*, 2005, pp. 395–402.

[9] M. Pichler and H. Rumetshofer, "Business process-based requirements modeling and management," in *Proceedings of the First International workshop on Requirements Engineering Vizualization (REV06), September 11, Minneapolis, USA*, 2006.

[10] *Rational Unified Process*, IBM. [Online]. Available: http://www-01.ibm.com/software/awdtools/rup/

[11] *Office Open XML*, ISO.