Criteria Used in Selecting Effective Requirements Elicitation Procedures

Abstract: New methods, methodologies and procedures are often introduced within information systems without discussing the effectiveness thereof. We investigate the effectiveness of a requirements elicitation procedure for deriving the higher education process model structures. There were three activities that 'rig' this study before comments were possible on the effectiveness of the requirements elicitation procedure. Firstly, the characteristics that the procedure adhered to were identified from an existing list of characteristics, which enables us to report on the characteristics of the requirements elicitation procedure developed for higher education institutions. Secondly, the procedure was used at different institutions to investigate if the procedure was useful at more than one institution. Lastly, the process model structures produced by the requirements elicitation procedure were used in a reengineering activity, in order to comment on the usefulness of the process models produced by the procedure. After using the characteristics defined for an effective requirements elicitation procedure, it was possible to comment that the requirements elicitation procedure adhere to a significant number of characteristics, using the procedure in a HEI derives the desired process model structures and the process model structures produced are useful in activities such as reengineering.

1 Introduction

A process model is a structure that represents a group of processes and their relationship to one another, which together accomplish a specific goal. A high-level process model is defined as the structure depicting all the primary processes and their relation to one another to accomplish the high-level objectives of the modelling exercise (Van der Merwe, Pretorius et al. 2004).

Process model structures are effectively used in different application domains. In software development the development team uses process model structures to model the different software processes and the interaction between them (Davenport 1993; Hammer and Chamy 1994; Kawalek and Kueng 1997; Somerville 2000; 2000; Pressman 2005). In business applications process model structures are useful in reengineering of business processes where the goal is to enhance a single process or a number of processes (Lin, Yang et al. 2002). A business process can be defined as a set of activities designed with the aim of producing a specific output for a specific customer or market. A business process is therefore structure for action consisting of a specific ordering of work activities, with a beginning and an end, with clearly defined inputs and outputs: a structure for action (Sparks 2000).

In the context of enterprise modelling a process model is a description of ‘what the enterprise does’ and can be used to manage the enterprise and changes in the enterprise (Rolstadas and Andersen 2000). Enterprise process modelling is the first building block in aligning strategic objectives of the enterprise with information technology (IT) and other initiatives. The process model communicates to the enterprise management and other stakeholders, such as the IT department, how the objectives of the enterprise are fulfilled through the processes of the enterprise. An enterprise process can therefore be defined as a logical chain of events that uses an enterprise’s resources to convert an object for the purpose of achieving some specified and measurable result/product.

All these process models have one thing in common though: they must all first be established before they can be of value. The effectiveness of the requirements elicitation procedures to establish such structures is the main topic of this paper. We used the higher education institution (HEI) as enterprise application domain as case study environment, where worldwide there currently are a number of reengineering activities, due to the introduction of technological innovations caused by new technologies such as the Internet and World Wide Web (Bates 2003; Lazenby 2003). This motivates the relevancy of selecting the HEI as application domain.

The primary contribution of the paper is a set of criteria that can be used as guidelines to establish whether a proposed requirements elicitation procedure for the establishment of process model structures is effective, and therefore reusable.

1.1 Focus of this paper

The problem that the project manager in an HEI therefore faces, in considering the adoption of the proposed requirements elicitation procedure for his/her HEI, is ‘how do I know that the elicitation procedure that I am considering will be effective and may be successfully used in my HEI?’.

The danger in introducing a new procedure is that researchers may fall into the trap of only developing the procedure, use it at one enterprise, and then publish the results as if it can be generalized. Although there is a probability that such a procedure will be successfully reused, it can be a very expensive exercise if the necessary outcome is not reached.
In a short survey conducted a number of project managers were asked how they select a requirements elicitation procedure. The majority of respondents replied that they use existing popular requirements elicitation procedures, without any formal selection method on the effectiveness of the procedure. This may lead to unsatisfactory results, and it is therefore necessary to investigate a more formal method as to the selection of the procedure. We argue that a manager should consider properties such as the reusability of the procedure and also the usefulness of structures produced by the procedure. We encapsulated these characteristics in the term “a effective requirements elicitation procedure”, and argue that a effective requirement elicitation procedure is a procedure that:

- adheres to requirements elicitation procedure characteristics;
- produces the desired process model structures in more than one enterprise; and
- produces process model structures that are useful in activities such as developing new systems and reengineering.

In order to comment on the appropriateness of these criteria, we used the criteria within the HEI application domain to analyse and comment on the effectiveness of an existing requirements elicitation procedure developed for the HEI domain.

Although the criteria defining a effective requirements elicitation procedure was defined and tested within the HEI domain, the significance of this work is that the criteria are generic and can therefore be used to assess other requirements elicitation procedures aimed at determining process model structures as well.

1.2 Layout of the paper

We first give an overview of a requirements elicitation procedure developed for requirements elicitation in HEI institutions, which is used in further discussions. The requirements elicitation procedure is then evaluated against a set of characteristics for elicitation procedures, the output produced in different environments are discussed and the process model structures produced at one institution is used in a reengineering activity. This is followed by a brief discussion on the effectiveness of the requirements elicitation procedure and concluding remarks.

2 The requirements elicitation procedure for HEIs

In this section we give a short overview of the phases for the suggested requirements elicitation procedure for establishing the high-level process model for HEIs. The procedure, called REPPMS (Requirements Elicitation Procedure for Process Model Structures), was first published in Van der Merwe et al. (2004) and later refined and published in (Van der Merwe 2005) (for a detailed discussion on the application of the procedure the interested reader is referred to these publications).

Process modelling presents a technique (involving several activities) to graphically depict the series of processes that accomplish a predefined goal (Curtis, Kellner et al. 1992; Snowdown 2002). REPPMS focuses on the identification of both the high-level process model that includes the primary processes of the HEI institution as well as the sub-process models that constitute these primary processes.

Porter (1985) developed the concept of a value chain in which he distinguishes between the primary and secondary process. For the purpose of this study, primary processes are those critical activities responsible for (or involved in) the design and construction of the student’s learning environment. Support processes are those processes that provide sustenance for the primary processes playing a secondary role in accomplishing the defined goal, for example human resources, catering services, etc. For a specific application domain, there is only one high-level process model consisting of the primary processes, with possibly several smaller sub-process models to augment and refine the high-level process model. Therefore, the procedure involves not only the activities to create a high-level process model, but also the essential sub-process models.

There are a number of significant elements that are used to depict a particular process, and different process modelling methodologies suggest different significant elements all depending on the specific application domain. Wang (1999) describes different elements for a process model, including an activity, a task, input/output, roles and a user. Eriksson and Penker (2000) provide a higher abstract level of these elements to include the process itself, process resources and the goal description of the process. Process resources can either be input or output resources. An input resource is used to assist in the flow of process activities. For example, in a student registration process, the registration form (input) is used (initially) to capture the student information. An output resource is the resulting output of the activities in a specific process, and in turn might serve potentially as an input resource to another process. Each process has at least one input resource and one output resource associated with it.

REPPMS was formalised as a requirements elicitation procedure in the form of a spiral model with five different phases (Figure 1), where each phase may be revisited more than once in order to accomplish the desired results.
2.1 Phase 1: Establish high-level objectives

In Phase 1, the requirements engineering team, in cooperation with stakeholders, compiles a detailed description of the higher-level purpose of the requirements elicitation exercise. The higher-level purpose focuses on approval for the adoption and integration of new systems affecting the entire organisation, and the stakeholders at this stage usually comprise members of the management of the institution. If management does not launch the requirements elicitation initiative, it is at least essential that approval and collaboration commitment be secured before continuation. This is necessary because one of the primary causes of unsuccessful or rejected projects is the failure to establish upper-management commitment to these projects (Singh 2000; Whitten, Bentley et al. 2000).

The deliverable of the first phase is a descriptive document acting as a framework available for future reference and verification purposes. A document of this nature includes a short description of the objective(s) as well as a clear specification of the required deliverables. Typically, it includes a single primary objective supported by one or more secondary objectives. A primary objective rationalises the reason for performing the requirements elicitation exercise, acting as guidance throughout the elicitation exercise and also during the development and deployment of the intended systems. A lack of awareness of the primary goal might cause the requirements engineering team to deviate from their task unnecessarily, leading to expensive time delays. The secondary goals serve as a refinement of the primary goal and often also embody constraints within the application domain.

2.2 Phase 2: Identify critical institutional units

In Phase 2, the goal is to identify the different critical units at the institution. As a first step, all the units at the institution are listed — this can be done by retrieving information from documentation and diagrams, such as organisational charts, or through interviews. The second step involves extracting those units that are actively involved in the creation and presentation of learning environments. Units focusing on other aspects of the institution are then labelled as support units and are deleted from the unit list. For example, the development team may feel that the catering services unit, which prepares refreshments, is not involved in the establishment of a learning environment, and will therefore be removed from the unit list.

The deliverable of Phase 2 is a listing of the critical operational units of an institution.

2.3 Phase 3: Identify primary processes

The purpose of Phase 3 is to identify the primary processes in each of the critical units of the application domain.

In the case of small institutions, the identification of core processes and follow-up results is generally simple, but the complexity often increases dramatically with the size of an institution. In the next three phases, we therefore used mathematical notation to define the different processes and resources. Using mathematical notation to describe a specification provides developers with the means to:

- present the detail accurately and concisely;
- express the interpretation assigned to specific aspects unequivocally;
- make the different results portable, reusable and extensible; and
be both operational and expressive (Kotze and Cloete 2004). This will allow for the scalability of the REPPMS for a variety of institution sizes and complexity.

The Process Model Inc. (2001) suggests that identification of primary or core processes is a first step towards constructing a process model. Porter (1985) identifies five primary activities in the business environment which contribute to the value of businesses. The activities include inbound logistics, operations, outbound logistics, marketing and sales, and services. Applying the fundamentals of his work to the HEI application domain yielded a list of primary processes applicable to this domain. This list should be considered only as a starting point since modifications or expansion might be necessary to describe the application domain correctly and completely. The elements of the starting list include:

1. The registration process (REGISTRATION).
2. Development of course material (COURSE DEVELOPMENT).
3. Production (PRODUCTION) of course material. At face-to-face teaching (also sometimes called residential) universities, this activity is often embedded in the development of course material, and is the responsibility of lecturers. At distance learning institutions, it is a separate process, which is handled by sections responsible for production of the material.
4. Distribution of course material (DISTRIBUTION).
5. Academic support available to the student (ACADEMIC STUDENT SUPPORT).

The following steps can be used to expand the above list and to verify its adequacy and completeness. These steps should be applied to the unit list created in Phase 2, and repeated for each of the units.

1. List and document the most important processes of the particular unit in order to establish the main duties within it. The focus is on the goals to be achieved rather than on the individual activities that might realise these goals. A general guideline is to include what-processes rather than how-processes. A what-process is goal-oriented in its description, expressing the objective of the particular process, while a how-process is action-oriented, explaining the particulars of specific activities to accomplish the specified goal.
2. Categorise each process as either being a support or a primary process using the definitions provided above.
3. Attempt a mapping of each of the newly identified primary processes to an item on the starting list of processes. A process list is created from items on the starting list that correspond to primary processes through their mappings, whilst primary processes that cannot be mapped are added as new items on the process list.

The deliverable of Phase 3 is a process list consisting of a set of the identified primary processes ($P$), namely

$$\{P_k\}_{k=1}^m$$

where $m$ denotes the total number of processes for all critical operational units of the HEI.

Eriksson and Penker (2000) comment that it is unusual, even for a complex environment, to have more than ten primary processes and they advise modellers to identify only between five and ten primary processes portraying the high-level duties that add value to an organisation. In the case of more than ten processes, it is advisable for the development team to reconsider individual items on the process list and, where possible, combine items with close associations. A model with too many processes is complex to interpret and as a result loses some of its effectiveness intended to improve understanding.

### Phase 4: Construct the high-level process model

The first construction step towards the high-level process model is to define the goal, input resources and output resources associated with each item on the process listing created in Phase 3. At the end of this step, a set of all the resources ($R$) for primary processes of the application domain can be described as:

$$\{R_j\}_{j=1}^n$$

where $n$ is the total number of resources.

Furthermore, the set of all goals ($G$) for the primary processes are defined as:

$$\{G_i\}_{i=1}^m$$

where $m$ is the total number of processes for the institution. Each goal is associated with one process and the total number of goals is therefore equal to the total number of processes.

The second step is to indicate the workflow between the different primary processes through input and output resources. The objective is to identify the resources that serve as either input or output resource, or both, for the different processes and then to eliminate redundant resources.

This task remains simple as long as there is only a small number of primary processes to consider and can be done by simply connecting related processes through directed lines. However, as the number of primary processes increases, the complexity of depicting the workflow also increases considerably. In the latter case a more formal approach is suggested to establish relationships between primary processes.
To identify these resources, determine the association value \( A \) that a resource \( R_j \) has with a process \( P_k \) (for all \( j \) and all \( k \)). The association value \( A \) may be input \((I)\), output \((O)\), or no association \((null)\). Each value is stored as an entry in an association list, which tabulates vertically all processes from top to bottom, and horizontally all resources from left to right. At the end of this process the set of processes and the resources associated with the various processes is described as:

\[
\forall \{P_k\}_{k=1}^m \land \forall \{R_j\}_{j=1}^n \bullet \exists AT_{kj} = (P_k, R_j, A)
\]

where \( A \in \{I, O, null\} \).

The following steps are followed to indicate the workflow and associations between the different processes on the process model diagram:

1. Construct the association table \( AT_{kj} \): For \( k=1..m \) and \( j=1..n \), describe all the resources \((R_j)\) in terms of their association values \( A \) with \( P_k \).
2. For \( k=1..m \), graphically depict \( P_k \) on a diagram with its associated goal \( G_k \).
3. For \( j=1..n \), add the identified resources, \( R_j \) to the diagram.
4. Use the set of triples \( AT_{kj} \), in particular the third coordinate, \( A \), to add directed lines between processes and resources: for an input \((I)\) association an \( \rightarrow \) is added from the resource to the process; for an output \((O)\) association an \( \rightarrow \) is added from the process to the resource. \( Null \) values are ignored, i.e. not indicated on the process model diagram.

The deliverable of Phase 4 is the high-level process model for the first stage and the sub-process model for the refinement stages.

### 2.5 Phase 5: Refine the high-level processes and determine the sub-processes

As mentioned earlier, a complete understanding of the application domain is depicted through a single high-level process model with several smaller sub-process models to accomplish the intended goal. The purpose of the refinement phase is to decompose and particularise the individual processes in the high-level process model through iterative steps into a set of sub-processes or atomic activities. An atomic activity is a process that cannot be broken down into further sub-processes. The steps to derive the atomic activities (or sub-processes) are similar to those described in the previous phase for the high-level process diagram:

1. For each primary process \( P_k \) (which will be a sub-process during further refinement), identify the set of affiliated sub-processes \( P_{ki} \) where \( l=0..s \) where \( s \) denotes the total number of sub-processes involved with process \( P_k \) in the generation of its output resource(s).
   \[
   \forall P_k \bullet \exists \{P_{kl}\}_{l=0}^s \text{ with } l \in N
   \]
   where \( s \) denotes the total number of sub-processes involved with process \( P_k \) in the generation of its output resource(s).
2. For each sub-process \( P_{kl} \), define its associated goal, input and output resources as described in Phase 4. Associate the sub-processes with one another through input and output resources.
3. Draw the sub-process model diagram, which depicts the sub-processes, their relationships and goals graphically.
4. Repeat these steps for each of the identified sub-processes until all processes are atomic or unit the requirements engineering team decides against further refinement.

The deliverable of this step is a set of smaller sub-process models augmenting the high-level process model.

### 3 Determining the effectiveness of the requirements elicitation procedure

Best practices often drive the development of methodologies (Avison and Fitzgerald 2003). Different methods were used to find solutions to problems and later to document the ‘ideas’ and ‘best practices’. As soon as a solution was needed for a similar problem, the previous documentation was used as a guideline. These guidelines later evolved into methodologies that even became available commercially. There were no ‘recipes’ with guidelines on how to develop a procedure. We believe the reason is that procedures or methodologies are based on the documentation of best practices and there is not really a set of rules for creating them, except to say that they should produce the desired result rapidly, efficiently and cost-effectively (Frese and Sauter 2003).

When we were therefore faced with the problem of assessing REPPMS we again had no established procedure to follow and had to determine the procedure from scratch. In the development of the procedure we therefore investigated best practices from different development environments and integrated these to establish the criteria for assessing whether a requirements procedure is effective and therefore reusable.

As mentioned previously, we argue that a requirements elicitation procedure is effective, and therefore reusable, if it:
adheres to requirements elicitation procedure characteristics (section 3.1);
produces the desired process model structures in more than one enterprise (section 3.2); and
produces process model structures that are useful in activities such as developing new systems and reengineering (section 3.3).

In remainder of this section we discuss these three criteria and illustrate their use by applying it to REPPMS.

3.1 Characteristics of a requirements elicitation procedure

Requirements elicitation and process modelling exist within cyclic methodologies that have the aim of developing software, or reengineering current environments (Pressman 2000; Hickey and Davis 2003). A requirements elicitation procedure with the aim of producing process models for the HEI domain should therefore adhere to characteristics found in similar procedures.

To determine its validity the requirements elicitation procedure must be evaluated against a comprehensive list of characteristics defined for the evaluation of requirements elicitation procedures. For evaluation, we used a characteristics list compiled by Van der Merwe, Kotzé & Cronje (2005), which focuses on the requirements elicitation, modelling and cross-phase activities (Table 1).

Table 1: List of characteristics (Van der Merwe, Kotzé et al. 2005)

<table>
<thead>
<tr>
<th>Sub-phase</th>
<th>Characteristic</th>
</tr>
</thead>
<tbody>
<tr>
<td>Support</td>
<td>Provide automated support for the requirements elicitation process</td>
</tr>
<tr>
<td>Standards</td>
<td>Provide standardised ways of describing work products</td>
</tr>
<tr>
<td></td>
<td>The precision of definition of its notation</td>
</tr>
<tr>
<td></td>
<td>Process model standards</td>
</tr>
<tr>
<td>Techniques</td>
<td>Select appropriate technique for the problem domain</td>
</tr>
<tr>
<td></td>
<td>Use of use cases to describe related tasks</td>
</tr>
<tr>
<td></td>
<td>Support a systematic step-by-step approach</td>
</tr>
<tr>
<td></td>
<td>Modifiable solutions and be iterative in</td>
</tr>
<tr>
<td>Documentation</td>
<td>Support documentation of requirements</td>
</tr>
<tr>
<td>Maintenance</td>
<td>Procedures for maintaining work products</td>
</tr>
<tr>
<td>Conflict</td>
<td>Conflict negotiation</td>
</tr>
<tr>
<td>Specification</td>
<td>Requirement completeness</td>
</tr>
<tr>
<td></td>
<td>Requirement relevance</td>
</tr>
<tr>
<td></td>
<td>Expectations during specification of requirements</td>
</tr>
<tr>
<td></td>
<td>Correctness</td>
</tr>
<tr>
<td></td>
<td>Communication during specification of requirements</td>
</tr>
<tr>
<td></td>
<td>Requirement accuracy</td>
</tr>
<tr>
<td></td>
<td>Importance of necessity: requirements document</td>
</tr>
<tr>
<td></td>
<td>Level of control over specifying requirements</td>
</tr>
<tr>
<td>Boundaries</td>
<td>Specify constraints / boundaries</td>
</tr>
<tr>
<td>Problem analysis</td>
<td>Support analysis</td>
</tr>
<tr>
<td></td>
<td>Degree of understanding of the task and process</td>
</tr>
<tr>
<td>Data gathering</td>
<td>Support data-gathering techniques</td>
</tr>
<tr>
<td>Client/customer</td>
<td>Support customer/client involvement</td>
</tr>
<tr>
<td>Support modelling</td>
<td>Motivation to support modelling</td>
</tr>
<tr>
<td>Goal Modelling</td>
<td>Model the purpose by describing behaviour</td>
</tr>
<tr>
<td>User involvement</td>
<td>Reflect the needs of customers / users</td>
</tr>
<tr>
<td>Modelling</td>
<td>Model business rules</td>
</tr>
<tr>
<td></td>
<td>Support modelling of work flows</td>
</tr>
<tr>
<td></td>
<td>Clarity of business process</td>
</tr>
<tr>
<td></td>
<td>Model system services</td>
</tr>
<tr>
<td></td>
<td>Systems architecture modelling</td>
</tr>
</tbody>
</table>

Using the characteristics identified each phase of a requirements elicitation procedure were evaluated as either strongly correlating with the characteristic, partially correlating, or no correlation what so ever:
• Not adhere: the requirement elicitation does not adhere to the characteristic at all.
• Partially adhere: some aspects of the requirement elicitation adhere to the characteristic.
• Strongly adhere: the requirement elicitation procedure fully adheres to the characteristic.

3.1.1 Assessing REPPMS against the characteristics

The result of the rating of the different aspects of REPPMS is presented in Table 2.

<table>
<thead>
<tr>
<th>Sub-phase</th>
<th>Characteristic</th>
<th>Not Adhere (NA)</th>
<th>Partially Adhere (PA)</th>
<th>Strongly Adhere (SA)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Support</td>
<td>Provide automated support for the RE process</td>
<td>√</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Standards</td>
<td>Provide standardised ways of describing work products</td>
<td></td>
<td>√</td>
<td></td>
</tr>
<tr>
<td></td>
<td>The precision of definition of its notation</td>
<td></td>
<td>√</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Process model standards</td>
<td></td>
<td></td>
<td>√</td>
</tr>
<tr>
<td>Techniques</td>
<td>Select appropriate technique for the problem domain</td>
<td></td>
<td></td>
<td>√</td>
</tr>
<tr>
<td></td>
<td>Use of use cases to describe related tasks</td>
<td></td>
<td>√</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Support a systematic step-by-step approach</td>
<td></td>
<td>√</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Modifiable solutions and be iterative in nature</td>
<td></td>
<td></td>
<td>√</td>
</tr>
<tr>
<td>Documentation</td>
<td>Support documentation of requirements</td>
<td></td>
<td></td>
<td>√</td>
</tr>
<tr>
<td>Maintenance</td>
<td>Procedures for maintaining work products</td>
<td></td>
<td>√</td>
<td></td>
</tr>
<tr>
<td>Conflict</td>
<td>Conflict negotiation</td>
<td></td>
<td>√</td>
<td></td>
</tr>
<tr>
<td>Specification</td>
<td>Requirement completeness</td>
<td></td>
<td>√</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Requirement relevance</td>
<td></td>
<td>√</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Expectations during specification of requirements</td>
<td></td>
<td></td>
<td>√</td>
</tr>
<tr>
<td></td>
<td>Correctness</td>
<td></td>
<td>√</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Communication during specification of requirements</td>
<td></td>
<td></td>
<td>√</td>
</tr>
<tr>
<td></td>
<td>Requirement accuracy</td>
<td></td>
<td>√</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Importance of necessity: requirements document</td>
<td></td>
<td></td>
<td>√</td>
</tr>
<tr>
<td></td>
<td>Level of control over specifying requirements</td>
<td></td>
<td></td>
<td>√</td>
</tr>
<tr>
<td>Boundaries</td>
<td>Specify constraints / boundaries</td>
<td></td>
<td>√</td>
<td></td>
</tr>
<tr>
<td>Problem analysis</td>
<td>Support analysis</td>
<td></td>
<td>√</td>
<td></td>
</tr>
<tr>
<td>Data gathering</td>
<td>Support data gathering techniques</td>
<td></td>
<td>√</td>
<td></td>
</tr>
<tr>
<td>Client/customer</td>
<td>Support customer/client involvement</td>
<td></td>
<td>√</td>
<td></td>
</tr>
<tr>
<td>Support modelling</td>
<td>Motivation to Support modelling</td>
<td></td>
<td>√</td>
<td></td>
</tr>
<tr>
<td>Goal Modelling</td>
<td>Model the purpose by describing behaviour</td>
<td></td>
<td>√</td>
<td></td>
</tr>
<tr>
<td>User involvement</td>
<td>Reflect the needs of customers / users</td>
<td></td>
<td>√</td>
<td></td>
</tr>
<tr>
<td>Modelling</td>
<td>Model business rules</td>
<td></td>
<td>√</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Support modelling of workflows</td>
<td></td>
<td>√</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Clarity of business process</td>
<td></td>
<td>√</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Model system services</td>
<td></td>
<td>√</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Systems architecture modelling</td>
<td></td>
<td>√</td>
<td></td>
</tr>
</tbody>
</table>

REPPMS strongly adhered to the use of standard notation and existing process model standards. It also supports a step-by-step approach which is defined in the original documentation to be iterative. More than once in the REPPMS it refers to the output of a phase being a set of documentation; it therefore, also supports the use of documentation of the requirements. REPPMS supports requirement relevance by excluding units and processes that are not applicable to the goal, modelling the primary processes that are important in creating a learning environment. This goal and the limitations are expected in the beginning of REPPMS, which indicates that the developers support the definition of expectations and the specification of boundaries. REPPMS suggested a systematic procedure to gather the necessary information from the different units, with the goal to gather correct, necessary and accurate information. REPPMS divides the educational environment into units for gathering
information, where the procedure uses communication techniques to extract necessary information from the employees.

The goal of REPPMS is to analyse the current environment so that a different developer can, with this information and his understanding of the environment, identify tasks and processes within the educational domain. Three of the five phases in REPPMS are concerned with the modelling task. REPPMS therefore strongly adheres to the modelling of business rules, the workflows and different services. REPPMS gives a motivation for using modelling in this application domain and also adheres to the purpose by producing the goal, the high-level process model and sub-process models.

There are only a small number of characteristics that REPPMS does ‘not adhere to’. One characteristic that needs further investigation is the automated support for the requirements engineering process. As mentioned, it should be possible to use existing tools to support the documentation process, such as existing case tools. REPPMS suggested only one way of gathering information, while other techniques such as questionnaires could also be appropriate for the application domain. Furthermore, REPPMS did not specifically mention the importance of maintenance, but support the use of documentation, which is easily maintainable. REPPMS does not address conflict negotiation, which is an important characteristic, and research into this is necessary, especially in the educational domain with diverse personnel involved in development. Although REPPMS do not specifically define measurements to measure requirements completeness, it does suggest a cyclic system, which tries to get complete requirements. Because the goal of REPPMS is to model the current enterprise processes, there is no need analysis involved. If REPPMS was to be used in the development of new systems, this issue will have to be addressed. Finally, REPPMS does not include system architecture modelling, which is an important aspect during the redesign of workflows.

3.1.2 Reflection on characterising REPPMS

From our investigation it was possible to summarise the strengths of REPPMS as follows:

- All the phases in the procedure support a systematic approach.
- The procedure is iterative in nature (the procedure is cyclic and is completed only after a number of iterations).
- In all phases, the information gathered by the developers is documented. This indicates that the procedure supports the documentation of the requirements and the documentation of the different models.
- In Phases 3 to 5, a notation used by modellers in process modelling environments is prescribed. The characteristic of providing standardised ways of describing work products is therefore adhered to.
- The notation is precise and process model standards are used.
- The characteristics that REPPMS did not adhere to should not change the outcome of the requirements elicitation procedure, but may be refined in future research to improve it.

It is therefore possible to claim that, based on the set of characteristics that the procedure does adhere to, REPPMS is an efficient requirements elicitation procedure which does adhere to the modelling and elicitation characteristics identified.

3.2 Producing the desired process model structures in more than one enterprise

In order for a requirements elicitation procedure to be deemed effective and reusable it must produce desirable results in more than one instance, i.e. it must be successfully used at more than one enterprise to model the process models relevant to the specific enterprise.

After the development of REPPMS it was used at the University of South Africa (UNISA) to test its applicability. UNISA is a distance teaching mega-university. In order to comment on the success and results obtained from a requirements elicitation procedure, it is, however, necessary to use the procedure in more than one enterprise and to consider the results at these various enterprises. In order to test REPPMS for various HEI models we selected two universities representing the other two educational models in South Africa: the University of Pretoria (UP) and the Tshwane University of Technology (TUT) for this activity. These two institutions both offer face-to-face teaching, in contrast to the distance education model used at UNISA, but they differ in basic institutional philosophy with UP being a ‘traditional’ university, while TUT also focuses on vocational training. Furthermore, both institutions were accessible, being near to UNISA, and management at both these institutions agreed to the data gathering at these institutions.

For data collection we prepared interview sheets, identified documents that could assist in the elicitation, did some context analysis on documents from the institutions and used the institution’s web pages to acquire information on the different units. First the different units were identified with contact persons in each unit. Most of the interviews were conducted telephonically with the different departments. The units, for which it was not possible to apply pre-knowledge, were visited to take field notes or to conduct interviews to gain an understanding of the working of the unit at the specific university.
After the elicitation activities, the process models were constructed from the data gathered at each institution. We briefly discuss the activities at each of the three institutions below.

3.2.1 Testing the high-level process model structure at UNISA

REPPMS was first used at the UNISA to establish the process model structures at the institution. The higher-level process model established after Phase 4 of REPPMS is illustrated in Figure 2.

In this representation of the HEI structure, the flow between processes is supported by fourteen resources. The input for the first process, REFLECTIVE RESEARCH, is the research material used to conduct the research. This includes prescribed books, journals, publications, web-based resources, etc. The output for this procedure is a staff member who can be seen as knowledgeable on the research topic and/or a written report on the findings of the research activity. Both these can be input resources for COURSE DEVELOPMENT where the output is a piece or pieces of study material, including tutorial letters, study guides, examination papers, video, audio etc. These source documents needed for duplication are sent to the PRODUCTION process where the printing/production is started based on the number of students in the course (retrieved from STUDENT SYSTEM). The DISTRIBUTION process sends course material to students based on student information retrieved from STUDENT SYSTEM. Material could also be distributed from other resources, e.g. from the library (books). REGISTRATION is done using an application form received from the student, his/her academic record and the rules of the institution for registration. The data is captured and stored on the STUDENT SYSTEM. ASSESSMENT is done based on the assignment/examination paper received from the students (once again the student information is retrieved from STUDENT SYSTEM). For ACADEMIC STUDENT SUPPORT the lecturer needs the student information (if it is relevant to marks obtained), the course material (if it is course related) and/or the assessment results to assist successfully in answering queries.

REPPMS was therefore deemed successful in modelling of the processes of a typical distance HEI.

3.2.2 Reusing REPPMS at the University of Pretoria

The first four phases were followed at UP and the high-level process model was derived successfully. For Phase 5, the refinement of the high-level process model, the REGISTRATION process was selected for decomposition. Figure 3 illustrates the second-level process model structure for the electronic registration process.
Figure 3: Second-level process model structure at UP

The resulting process model structures were discussed with representatives at UP and small changes were made to lower level representations. The representatives agreed that the structures were a representation of the processes at the time (August 2002) and that it may be a valuable tool for communication between members of staff, in activities such as reengineering.

3.2.3 Reusing REPPMS at the Tshwane University of Technology

For the application of REPPMS at TUT we constructed the high-level process model structure and, similarly to UP, also refined the REGISTRATION process. The requirements elicitation procedure also proved to be effective in delivering the process model structures at this institution, where after application process models were delivered that was described by staff members as useful and representative of the institution structures.

After two iterations of the procedure it was much easier to use it at the third institution. As was the case at UP, there were some negative respondents who were unsure about the purpose of the research and therefore questioned the reason for the interviews. This highlighted the fact that negotiation skills, including conflict negotiations, should form part of the process modeller’s skills.

3.2.4 Reflection on the using REPPMS at various enterprises

REPPMS was successfully tested at UNISA and then reused at UP and TUT to model the existing high-level processes. The lower-level process model structures for the REGISTRATION process were also modelled. Although there were notable differences in the lower-level processes, the high-level process models for all three institutions were very similar, leading us to the believe that a generic high-level process model for HEIs can be determined.

REPPMS thus satisfied the criterion of being able to produce desired process model structures at various enterprises.

3.3 Producing process model structures that are useful in design or reengineering activities

The last concern in the list of requirements for the requirements elicitation procedure is that the output should be useful in an activity such as designing a new system or reengineering.

According to Hammer (1990), there are five steps in reengineering when using a process model:

1. Name the processes and state your goal.
2. Map the process.
3. Choose the process to reengineer.
4. Understand each process.
5. Reengineer the process.

These five steps correlates with the five steps defined for process reengineering by Davenport and Short (1990). In both approaches the goal is to identify the processes, to focus on the process to be re-engineered and to understand the process (Omiri 2000).
Another approach is one suggested by Van der Merwe, Kotze and Cronje (2006), where the focus is on delays between processes. In this approach, theory of constraints (TOC) as introduced by Goldratt and Cox (1992) was selected as the basis for identification of constraints within the process model. The procedure consists of five phases, including the identification of the process with the constraint, identification of the constraint in the sub-process, the reason for the constraint, the solutions for the problem and the implementation of the solution, as illustrated in Figure 4.

![Figure 4: A reengineering procedure based on TOC (Van der Merwe, Kotzé et al. 2006)](image)

To identify the process with a constraint the following steps are followed:
1. Use a high-level process model to identify (or focus on) possible constraints.
2. Derive from the process model a table that lists all the processes \( P_k \) where \( k = 1..m \), where \( m \) denotes the total number of processes.
3. List a throughput value and a demand value for each process. The possible values for throughput are the set \{possibility, none, satisfactory, a\} and similarly for demand the set \{possibility, none, satisfactory, b\}. A numeric value \( a \) or \( b \) is assigned to the attributes throughput and demand respectively, where it is possible to determine actual numeric values. A value of ‘possibility’ is assigned to the attributes throughput and demand if the re-engineering team suspects a constraint in sub-processes, but is not sure. A value of ‘satisfactory’ is to the attributes throughput and demand if the current throughput or demand is satisfactory and ‘none’ if it is not quantifiable.
4. Add a column indicating constraint values to the table with a value of ‘Yes’ indicating a constraint or a value of ‘No’ if not. This value is determined using the following algorithm assigning a ‘Yes’ value to any process in which the current throughput is less than the demand, or where the ‘possibility’ of a constraint exists:
   
   \[
   \begin{align*}
   & \text{if (throughput = ‘satisfactory’ or throughput = ‘none’)} \quad \text{then constraint = ‘No’} \\
   & \text{else if throughput = ‘possibility’ then constraint = ‘Yes’} \\
   & \text{else if demand > throughput then constraint = ‘Yes’} \\
   & \text{else constraint = ‘No’;} \\
   \end{align*}
   \]

   In Phase 2 a similar procedure is necessary for identifying the constraints in the selected sub-process:
1. Determine the list of sub-processes for the process being scrutinized.
2. Determine the demand and throughput values for each sub-process.
3. Identify the constraint(s) in the list of sub-processes as in Phase 1.
4. Select the sub-process(es) to be scrutinized.
5. If the selected sub-process has sub-processes, go back to Step 2 and repeat the procedure.

The deliverable of this procedure is a list of sub-processes for a process \( P_k \) on the higher-level process model, with one or more possible constraints within the list of sub-processes. The re-engineering team then decides on the biggest constraint that should be addressed in the remaining phases, or repeat the process for more than one.

3.3.1 Using REPPMS in a reengineering exercise at UNISA

We selected UNISA as the case study environment for reengineering because of the accessibility to data. UNISA encapsulates all the processes available at residential institutions, which makes it an ideal case study environment. We followed the reengineering procedure proposed by Van der Merwe et al. (2006) in this reengineering exercise.

For data gathering with regard to constraints in the reengineering activity, key representatives were identified and were contacted to discuss the possible constraints in the applicable departments. Data gathering included the
use of interviews, which were scheduled initially with one representative in the first phase of the reengineering procedure and in Phase 2 with representatives in activities where constraints were experienced. For data on constraints experienced within the registration process, SQL queries were used to obtain data on the throughput experienced during the registration period. In reports on delays in the course development, academic peers were consulted and the information was documented. The information in interviews was documented using interview templates for later reference purposes. Where possible, data was verified for validity against information from other resources.

We briefly discuss the various phases of the reengineering exercise below.

3.3.1.1 Phase 1: Identify the processes at UNISA subject to constraints

The five steps defined for Phase 1 in the reengineering procedure by Van der Merwe et al. (2006) were followed. The high-level process model identified previously (Figure 4) were used and according to the throughput and demand calculations, four processes were identified with possible constraints: COURSE DEVELOPMENT, PRODUCTION, ASSESSMENT and REGISTRATION (Table 3). The REGISTRATION process was selected by the stakeholders for the remaining steps of the reengineering process.

<table>
<thead>
<tr>
<th>Process</th>
<th>Throughput</th>
<th>Demand</th>
<th>Constraint</th>
</tr>
</thead>
<tbody>
<tr>
<td>P1</td>
<td>REFLECTIVE RESEARCH</td>
<td>None</td>
<td>None</td>
</tr>
<tr>
<td>P2</td>
<td>COURSE DEVELOPMENT</td>
<td>Possibility</td>
<td>Possibility</td>
</tr>
<tr>
<td>P3</td>
<td>REGISTRATION</td>
<td>71246 (1/12-9/2)</td>
<td>90739</td>
</tr>
<tr>
<td>P4</td>
<td>PRODUCTION</td>
<td>Possibility</td>
<td>Possibility</td>
</tr>
<tr>
<td>P5</td>
<td>DISTRIBUTION</td>
<td>Satisfactory</td>
<td>Satisfactory</td>
</tr>
<tr>
<td>P6</td>
<td>LEARNING ACTIVITIES</td>
<td>Satisfactory</td>
<td>Satisfactory</td>
</tr>
<tr>
<td>P7</td>
<td>ASSESSMENT</td>
<td>Possibility</td>
<td>Possibility</td>
</tr>
<tr>
<td>P8</td>
<td>ACADEMIC STUDENT SUPPORT</td>
<td>Satisfactory</td>
<td>Satisfactory</td>
</tr>
</tbody>
</table>

3.3.1.2 Phase 2: Identify constraints in sub-process

For the REGISTRATION process, four sub-processes were identified on the second level, including Application process, Academic verification, Payment verification and Course material distribution. The Application process was identified as the process with the throughput problem. For the Application process there was once again a list of ten sub-processes on the third level, including:
- Student number application
- Student number allocation
- Send confirmation of actions to student.
- Application form completion
- Put on work flow
- Course profile verification
- Course data capture.
- Send confirmation of actions to student.
- Register and verify student payment.
- Course material distribution

Student number application, Student number allocation, Register and verify student payment, Course profile verification and Course data capture were identified as possible constraint processes on the third decomposition level. The last two (Course profile verification and Course data capture) were selected as the sub-processes on which to focus because the biggest time delay was experienced in them (at the Undergraduate Unit nearly 10 000 student enrolments are delayed in the Course-Profile-Verification and Course-Data-Capture sub-processes).

3.3.1.3 Phase 3: Identification of reason(s) for a specific constraint

According to staff at the Undergraduate Unit, applications are in a queue (in the order received) where a first-in-first-out rule is applied. The physical processing of one application is more or less 10 minutes. During the interview the following were significant reasons given for the delays, including: staff members are constantly busy with telephone enquiries on the status of student applications; student applications are duplicated for fear that the first application has not been received; incorrect information is received from students, i.e. re-registration is required; there are only a few people who can handle the exceptions in course verification; the data on the expert systems used is not updated by responsible role players; management does not realise how dire the lack of resources is; counter students (65 000) involved in the REGISTRATION process get precedence over electronic/postal students, and in busy registration periods staff members are primarily assigned to the handling of counter registration (students that walk in receive priority).

It is preferable that the proposed solution should consider and address a large proportion of these concerns if it is to be considered successful.
3.3.1.4 Phase 4: Consideration of solutions to the problem

There is more than one solution for the electronic registration system. Finding a feasible solution for an electronic registration system at a university is a tedious task. The development team may consider the use of existing software that is available or they may decide to develop in-house software. The first option may seem ideal, but software available for administrative tasks of this nature is very expensive and it is often not possible to customise it to interact with existing systems. An alternative is to develop the system in the institution itself. This could also be an expensive option, but has the advantage that the software is customised according to the existing legacy systems and that costs are spread over a longer period, in contrast to buying which is an immediate expense.

A feasibility study is necessary, and because the purpose of this research was to focus on how one can use the process models, we only conducted a feasibility study into the option of changing existing systems.

At UNISA, the constraint that is a concern in the electronic registration system is in the Application Process on the second level. This sub-process is ideal for automation if there is a system that handles the application electronically. A system of this nature will be ideal if it can be a registration management system (RMS) that handles the application from inception until the final registration of the student. It will therefore not only benefit the Academic verification sub-process, but will also focus on the other constraints in the Application process, Payment verification and Course material distribution. This is in accordance with the reengineering procedure, which states that a solution can either focus on a single constraint at a time, or on a chain of events.

In the Application Process, we suggested the use of an RMS system similar to the one already in use at UP. The proposed solution is graphically depicted in Figure 5.

![Figure 5: The suggested Registration Management System](image)

For the Academic verification we suggested the use of the existing UNISA Expert System, but recommend that it be integrated with the central management system. For the Payment verification we recommend that the process makes provision for automatically registered payments. For the Course material distribution we suggested the use of a system where the student gains access to his/her course material as downloadable PDF material. In the centre of the suggested automated electronic system is the RMS, which is a software management system responsible for managing the application from the moment that the student initiates the application process until the course material is dispatched to the student. The activities managed by the RMS and the advantages of its implementation are described in detail in Van der Merwe (2005).

3.3.1.5 Phase 5: Implementation of changes and evaluation of results

For feasibility purposes, a small-scale version of the registration system was implemented by a project student at UNISA (Green and Van der Merwe 2004). Full implementation requires a long-term commitment from all stakeholders in the enterprise. The recommendations in this research may act as starting point in development. Some of the functions are already available independently at other institutions (such as UP) and therefore feasible for implementation.

3.3.2 Reflection on the reengineering exercise

To conclude we briefly reflect on the usefulness of REPPMS in the reengineering exercise by commenting on the role of process models in the exercise. Our finding is summarised in Table 4.
Table 4: Role of process models in different phases

<table>
<thead>
<tr>
<th>Phases</th>
<th>Documentation</th>
<th>Comments on the role of the process models</th>
<th>Indication of usefulness</th>
</tr>
</thead>
<tbody>
<tr>
<td>Phase 1: Identification of main constraint</td>
<td>High-level process model Process list</td>
<td>The high-level process model is used to identify the process list, which is then once again used to determine the constraint in each process. Without knowing what the processes are, it is impossible to identify the high-level constraint.</td>
<td>High</td>
</tr>
<tr>
<td>Phase 2: Identification of constraint in sub-process</td>
<td>Sub-process models Sub-process list</td>
<td>The sub-levels are used to identify the process lists and the constraint on each level. Without knowledge of what the sub-processes are, it is impossible to identify the constraint on each level.</td>
<td>High</td>
</tr>
<tr>
<td>Phase 3: Identification of reason for constraint</td>
<td>Reasons for constraints</td>
<td>Although the process models are not prescribed directly as a tool in this phase, it may be a valuable graphical tool in discussions with role players in the institution to investigate the reasons for constraints.</td>
<td>Low</td>
</tr>
<tr>
<td>Phase 4: Consideration of solutions</td>
<td>Solution options Feasibility study Process models</td>
<td>The process list is used to look at alternative chains for a constraint chain of processes or at innovations to enhance the sub-process scrutinised.</td>
<td>High</td>
</tr>
<tr>
<td>Phase 5: Implementation of changes</td>
<td>Adapted process models</td>
<td>After implementation it is necessary to update the existing process models for future reference of the chain of events depicting the flow within an institution.</td>
<td>Medium</td>
</tr>
</tbody>
</table>

The process model and process lists derived from the process model are used on all levels of the suggested reengineering procedure. In three of the five phases a ‘high’ value is given to ‘the extent’ to which the process models were used. Phase 5 received a ‘medium’ value for use in the reengineering procedure, while only Phase 3 received a ‘low’ value. None of the phases received a ‘none’ value. Although these results cannot be generalised, they provide strong initial support for the notion that the process models are useful to a high extent if used in a reengineering activity, such as that described in this article. It is useful both for deriving the processes with constraints, and ideal for reengineering and as a graphical tool in the process.

4 Reflection on the effectiveness of the requirements elicitation procedure

The goal of this paper was to report on the effectiveness of the REPPMS procedure developed for HEIs, and on a more generic level to argue that requirements elicitation procedures should adhere to the criteria proposed for effectiveness.

After using the proposed criteria on the REPPMS procedure, it is possible to reflect that project managers at HEIs, may use the REPPMS procedure knowing that it adheres to a significant number of characteristics set out for requirements elicitation procedures, is reusable at different institutions and produces useful process model structures.

We therefore propose that before project managers accept a requirements elicitation procedure, they should first consider these three criteria. This may reduce risks in producing incomplete process model structures. In Figure 6 we propose a model that project managers may consider that encapsulate the three pre-requisites for a requirements elicitation procedure to be considered as a effective requirements elicitation procedure.
In future research, this model should be tested against other requirements elicitation procedures for enterprise, business and software modelling, before it may be considered comprehensive in more than one field.

5 Conclusions

This paper proposed a set of criteria that can be used to assess requirements elicitation procedures aimed at the development of process model structures. We applied the set of criteria by investigating the effectiveness of REPPMS, a requirements elicitation procedure that was developed to establish process model structures for HEIs. It was found that REPPMS sufficiently adheres to these criteria for it to be considered effective and reusable for the establishment of process models for HEIs.

We have not established the completeness of the set of criteria to determine the effectiveness of requirements elicitation procedures, but have shown that it can be used as starting point for determining such effectiveness. Determining the completeness of the criteria or suggesting additional criteria is a field for future research.

For further research in relation to REPPMS, we suggest the investigation into the generic nature REPPMS and how it can be adapted to make it usable in other enterprise domains than that of HEIs. Another field of research relate to the range of HEIs addressed by REPPMS currently: we have only used REPPMS at HEIs in South Africa and it was found to applicable for producing process models at the various types of HEIs that exist. Further research could determine it applicability to HEIs in other countries as well.

Bibliography


