The Functionality of a Requirements Elicitation Procedure Developed for Process Modelling within the Higher Education Application Domain

Alta van der Merwe
Paula Kotzé
Johannes Cronjé

Abstract

Although various application domains use requirements elicitation procedures to gather information and model the application domain, only a few guidelines mention the important characteristics that are essential for such procedures. This paper reports on identifying a set of characteristics for a functional requirements elicitation procedure within the higher education domain.

Introduction

The Internet is no longer a new technology in the higher education institution application domain. In a recent report published by Educause (Educause 2003) an increase in the number of institutions in the USA that use the Internet to provide web-based campus portals was reported to have risen from 21.2 percent in 2002 to 28.4 percent in 2003. Online registration facilities grew from 20.9 percent in 1998 to 70.9 percent in 2003. The same trend is noticeable in South Africa.

Traditional higher education institutions that have already incorporated e-learning into their curricula often claim to have gained an advantage over their competitors because they serve a larger number of students. If they wish to retain such an advantage, these institutions should continuously convert their existing processes so that they pro-

*Alter*nation 12.1b (2005) 489-514 *ISSN 1023-1757*

competitors do (Laurillard 1993; Bates 2000). However, the incorporation of e-learning into the curricula of traditional higher education institutions is a complex task (Luker 2000; Ryan, Scott et al. 2000). The main reason for dissatisfaction with e-learning and its often inefficient implementations is similar to those in most application domains where developments have not been successfully implemented (Pressman 2000; Whitten, Bentley et al. 2001). Reluctance to incorporate electronic innovations often originates in a failure to understand the application domain adequately. Such an understanding requires an expert appreciation of all aspects of e-learning technologies and strategies if it is to be successful.

Requirements elicitation is a technique used by organizations to describe and specify an application domain. Various requirements elicitation procedures are used to gather information and model environments in different application domains. In software development projects, for example, a number of software requirements engineering procedures are in use (McDermid 1993; Pressman 2000; Hickey and Davis 2003), and in the field of business process re-engineering, authors such as Davenport (1993) and Hammer (1990) describe specialized re-engineering elicitation procedures. In software and business environments numerous guidelines describe the characteristics of the procedures concerned. Although requirements elicitation procedures might therefore seem to be the ideal tools for building up an understanding of the higher educational domain, only a very limited number of descriptions of process modelling procedures for this environment exists, and likewise of the characteristics that a requirements elicitation procedure for this domain should adhere to (Bruno, Vrana et al. 1998; Tait 1999; Cloete, Van der Merwe et al. 2003).

The aim of our research is to gather information on the processes involved in creating a learning environment and in modelling the workflow between these processes. The objective of this paper is to identify the output characteristics of a functional requirements elicitation procedure applicable to the higher education domain. The identification of such a set of characteristics is especially beneficial to requirement elicitation procedure *developers* because it will help them to establish a procedure that includes all the important traits that such a procedure must possess. The recognition and inclusion of these vital traits will lead to the development of improved products in the course of re-engineering the current environment to include e-learning technologies. This would naturally increase the rate at which such products are successfully deployed and accepted. *Researchers* may also benefit from this paper because they are not only responsible for establishing and developing new knowledge that benefits society, but they work closely with practitioners to define much-needed standards.

The next section identifies the context of the paper with regard to the modelling of a complex environment, and elaborates on the procedure followed to establish the characteristics of the requirements elicitation and modelling procedures identified. This is followed by a section describing a requirements elicitation procedure that is applied in a higher education environment, and a section showing how this procedure adheres to the suggested characteristics. The penultimate section addresses the issue of the scientific validation of the reported research, while the final section makes a few concluding observations.

Identifying the Characteristics of the Requirements Elicitation and the Modelling of Processes

Modelling a complex environment, such as the changing educational domain, involves two main sub-fields, namely requirements elicitation and the modelling of the information gathered during the requirements elicitation process. *Requirements elicitation* is the systematic extraction and inventory of the requirements of a system (IEEE 1998). If a requirements elicitation procedure is to be considered effective, it should at least produce the initial goal (Rzepka 1989). *Process modelling* presents a technique (comprising several activities) that graphically depicts the series of processes that accomplish a predefined goal (Curtis, Kellner et al. 1992). The *process model* is the *structure* that represents a group of processes and their relationship to one another, which together accomplish a specific goal. These two sub-fields naturally exist within cyclic methodologies that have the aim of developing software or re-engineering current environments (Pressman 2000; Hickey and Davis 2003). Our focus, as illustrated in Figure 1, is on elicitation and modelling activities.

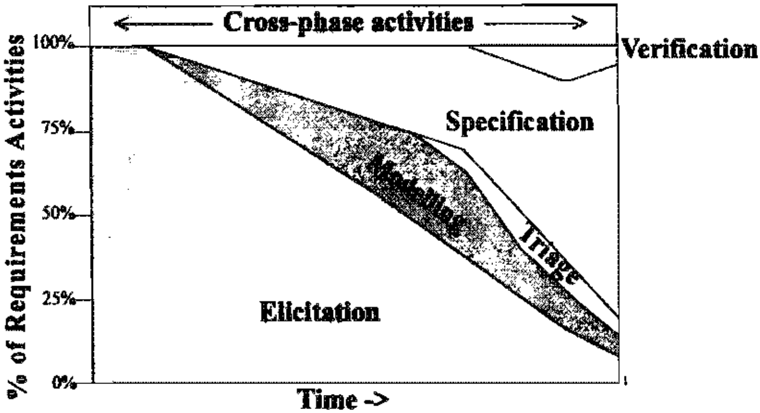

Figure 1: Requirements elicitation activities procedure (adapted from (Hickey and Davis 2003))

In order to identify the required characteristics of an elicitation procedure, we conducted a comprehensive literature review of sources that referred to characteristics within the field of study and described the characteristics of a range of features in requirements elicitation and modelling. Twenty-six of these resources mentioned useful characteristics. After a number of cycles of identification of characteristics and working through references, maturity occurred with fifty-eight identified characteristics.

Table A1 (Appendix A) shows a list of the twenty-six resources (a number has been assigned to each for further reference). Table A2 (Appendix A) specifies the list of characteristics that have been identified from these references, and includes a column with the corresponding references to a specific characteristic.

Although different authors propose different steps in the requirements engineering process, the core of these methodologies always includes (1) a feasibility study, (2) elicitation, (3) modelling, (4) triage, (5) verification and (6) cross-phase activities (Macaulay 1996; Sommerville and Sawyer 1997; Hickey and Davis 2003). We found that some of the characteristics identified as important for requirements elicitation actually belonged to other activities such as a feasibility stage – and were not relevant to an elicitation stage. By using these core steps and categorizing each of the characteristics into one of these steps, we ensured that we listed only those characteristics that were relevant to our study, namely the requirements elicitation and modelling phase (Appendix A, Table A2). We also merged characteristics with the same meaning and so ended up with a total number of fifty characteristics. In our last step, we grouped characteristics that naturally belong together into sub-phases (Appendix A, Table A2).

For the purpose of this paper, we are interested in only those characteristics that focus on the elicitation and modelling phases and in the ones that are applicable to all phases. Table 1 presents a list of these phases with the relevant characteristics identified. Although we appreciate the importance of the other phases, our focus in this paper will be limited to the phases mentioned.

A Requirements Elicitation Procedure for the Higher Education Environment

We now present an overview of a requirements elicitation procedure for the purpose of modelling a higher education environment. This procedure was developed and tested as part of a research project (Cloete, Van der Merwe et al. 2003) at the University of South Africa, and was also used as the fundamental requirements elicitation tool to determine the core and secondary processes at other institutions. The procedure consists of five phases. Phase 1 establishes objectives, whereas the identification of critical institutional units (Phase 2) and the identification of primary processes (Phase 3) help us to understand the domain. It is also during Phases 2 and 3 that the developers collect stakeholder requirements.

Table 1: List of characteristics

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

The procedure continues with the organization of the acquired information into a high-level process model (Phase 4), which is refined in the final step into several sub-process models (Phase 5). Each of these phases will now be described in more detail.

Phase 1: Establish High-Level Objectives

In Phase 1, the requirements engineering team, in cooperation with stakeholders, compiles a detailed description of the high-level purpose of the requirements elicitation exercise. The deliverable of the first phase is a descriptive document that acts as a framework for future reference and verification purposes. A document of this nature includes a short description of the goal(s) as well as a clear specification of the required deliverables.

Phase 2: Identify the Critical Institutional Units

The objective of the procedure is to identify the critical processes in the application domain (i.e. their essential activities and workflow) so that the application domain can be understood. The critical processes can only be identified from a consideration of different operational units within the institution. A *unit* refers to a working segment of the institution that is responsible for specific tasks such as, for example, a financial section, an academic department, a technical division, et cetera. As a first step, all such units within the institution are listed. This list is compiled from documentation and diagrams, such as organizational charts, or from data gathered using interviews. If e-learning is being contemplated, the second step involves extracting those units that actively create and present learning environments. The units that focus on other aspects of the institution are then labelled as *support units* and are deleted from the unit list. For example, while the catering services department prepares food, it has no direct responsibility the learning environment, and will therefore be removed from the unit list. The deliverable of Phase 2 is a list of the *critical* operational units of an institution.

Phase 3: Identify Primary Processes

In the next three phases, the procedure involves a formal approach to identify the relevant processes. The procedure distinguishes between primary and support processes in the application domain. Primary processes are the critical activities responsible for, or involved in, the design and construction of a student's learning environment. Support processes are those that provide sustenance for the primary processes and play a secondary role in accomplishing the defined goal. The purpose of Phase 3 is to identify the primary processes in each of the critical units of the application domain. The procedure suggests an initial list of primary processes for the e-learning domain that includes the registration process (REGISTRATION), development of course material (COURSE DEVELOPMENT), production of course material (PRODUCTION), distribution of course material (DISTRIBUTION), and academic support available to the student (ACADEMIC STUDENT SUPPORT).

The following steps will expand the list and verify its adequacy and completeness. These steps should be applied to each of the institutional units identified in the second phase (compiled into a unit list).

1. List and document the most important processes of a particular unit in order to establish the main duties within the unit. The focus should be on the goals to be achieved rather than on whatever individual activities might realize these goals. A general guideline is to include *what-processes* rather than *how-processes*. A *what-process* is goal-oriented in its description and expresses the objective of the particular process, while a *how-process* is action-oriented and explains the particulars of specific activities that will accomplish the specified goal.

2. Categorize each process as being either a support or a primary process by using the definitions provided earlier.

3. Attempt a mapping for each of the newly identified primary processes to an item on the initial list. A process list is created from items on the initial list that correspond to primary processes through their mappings, while primary processes that cannot be mapped are added to the process list as new items.

The deliverable of Phase 3 is a process list consisting of set of the identified primary processes, namely:

$$\{P_k\}_{k=1}^{m} \text{ for } k, m \in N_1$$

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

The procedure recommends that developers should reconsider the list if there are more than ten primary processes included in the list. Eriksson and Penker (2000) also comment that it is unusual, even for a complex environment, to have more than ten primary processes.

Phase 4: Construct the High-Level Process Model

The procedure for constructing a high-level process model employs a standard notation that includes the process itself, process resources and the goal description of the process (Eriksson and Penker 2000). Process resources can be either *input* (I) or *output* (O) 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 to capture primary information about potential students. An output resource is the resultant output of activities in a specific process. It might in turn serve as an input resource for another process. Each process has at least one input resource and one output resource that is associated with it. The first step towards constructing the high-level process model is to define the goal, input resources, and output resources that are associated with each item on the process list, which had been created in the previous phase. At the end of this step, a set of all resources for primary processes of the application domain can be described as:

$$\{R_j\}_{j=1}^{n} \text{ for } j, n \in N_1$$

with *n* denoting the total number of input and output resources.

The second step is to indicate the workflow between different primary processes through input and output resources. This task remains simple provided that (1) there are only a small number of primary processes to consider, and (2) they can be accomplished simply by connecting related processes through directed lines on the process model diagram. However, as the number of primary processes increases, the degree of complexity in depicting the workflow increases proportionately. In such cases, the procedure would indicate that a more formal approach to establish relationships between primary processes is required.

The objective is to identify the resources that serve as both input and output resource for the different processes and then eliminate redundant resources (those resources that would appear more than once on the same process model diagram). To identify these resources, determine the association value (say T_{kj}) that a resource R_j has with a process P_k (for all *j* and all *k*). These association values may be input (T_{kj} = I), output (T_{kj} = O), or *no* association (T_{kj} = Null). Each T_{kj} is stored as an entry in an association list, which tabulates vertically all processes from top to bottom and tabulates horizontally all resources from left to right.

The following steps assist in indicating the work flow and associations between the different processes and as a result describe the high-level process model:

1. For $k = 1..m$ and $j = 1..n$, describe all the resources in terms of their association values with P_k. This is written as a triplet (P_k, R_j, T_{kj}). (Null values can be ignored.)
2. For $k = 1..m$, graphically depict P_k on a diagram with its associated goal.
3. For $j = 1..n$, add the identified resources, R_j to the diagram.
4. Use the set of triplets (identified in 1), in particular the third coordinate, to add directed lines between processes and resources.

This approach produces a high-level process model for the application domain, as is illustrated in Figure 3.

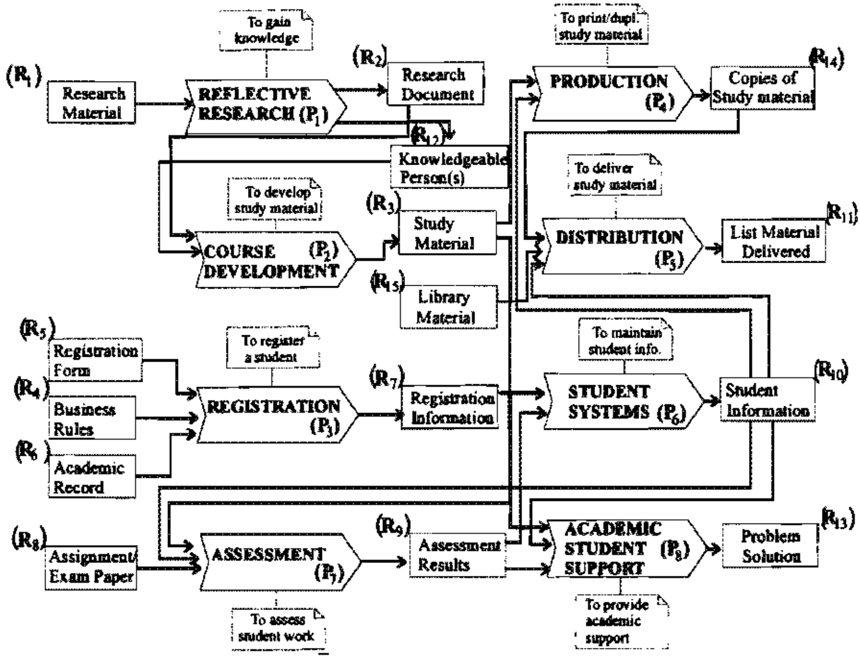

Figure 3: High-level process model for the higher education domain

Phase 5: Refinement

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 particularize 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 activities required to depict the sub-models mentioned above are similar to those described in the previous phase for the high-level process diagram. In summary:

1. For each (primary) process, identify the set of affiliated sub-processes involved in the generation of its output resource(s). For each sub-process, define its associated goal, input and output resources.
2. Associate the sub-processes with one another through input and output resources, as described in Phase 4.
3. Draw the process model. This model graphically depicts the sub-processes and their relationships between one another.
4. Repeat these steps for each of the identified sub-processes until all processes are atomic – or until the requirements engineering team decides against further refinement.

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

How Does the Requirements Elicitation Procedure Adhere to the Characteristics Identified?

In this section we consider the requirements elicitation procedure described above and portray its scientific soundness in terms of the characteristics listed in Table 1. Using the characteristics as a starting point, we evaluated and rated the requirements elicitation procedure with relation to each characteristic by using the following descriptors:

1. *Not adhere*: The requirements elicitation does not adhere to the characteristic at all.
2. *Partially adhere*: Some aspects of the requirements elicitation adhere to the characteristic.
3. *Strongly adhere*: The requirements elicitation procedure fully adheres to the characteristic.

The result of this rating of the different aspects of the requirements elicitation procedure is presented in Table 2. In the first column, we list the three phases followed by the sub-phases of each phase. In the third column, we include the characteristics identified followed by the rating achieved for each characteristic.

Table 2: Degree to which requirements elicitation procedure adheres to the identified characteristics

	Sub-phase	Characteristic	Does not adhere to (NA)	Partially adheres to (PA)	Strongly adheres to (SA)
All Phases	Support	Provide **automated support** for the requirements engineering process	√		
	Standards	Provide **standardised ways** of **describing** work products			√
		The **precision** of definition of its **notation**			√
		Process model standards			√
	Techniques	Select appropriate technique for the **problem domain**		√	
		Use of **use cases** to describe related **tasks**	√		
		Support a **systematic step-by-step approach**			√
		Solutions can easily be modified and are **iterative in nature**			√
	Documentation	Support **documentation of requirements**			√
	Maintenance	Procedures for **maintaining work products**		√	
	Conflict	**Conflict negotiation**	√		
Elicitation	Specification	**Requirement completeness**		√	
		Requirement relevance			√
		Expectations during specification of requirements			√
		Correctness			√
		Communication during specification of requirements			√
		Requirement accuracy			√
		Importance of **necessity**: requirements document			√
		Level of control over specifying requirements			√

	Boundaries	Specify **constraints / boundaries**			√
	Problem analysis	Support **analysis**			√
		Degree of understanding of the **task** and **process**			√
	Data gathering	Support **data gathering techniques**			√
	Client/customer	Support **customer/client involvement**			√
Modelling	Support modelling	Motivation to **support modelling**			√
	Goal modelling	Model the **purpose** by describing **behaviour**			√
	User involvement	Reflect the **needs of customers / users**		√	
	Modelling	**Model business rules**			√
		Support modelling of workflows			√
		Clarity of business process			√
		Model system **services**			√
		Systems **architecture modelling**	√		

The procedure 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 as *iterative.* Because reference is made more than once in the procedure to the output of a phase as being a set of documentation, it therefore also supports the use of *documentation* of the requirements.

Within the elicitation phase of the procedure, the procedure supports *requirement relevance* by excluding units and processes that are not applicable to the goal and modelling only the primary processes that are important in creating a learning environment. This goal and the limitations are expected in the beginning of the procedure. This indicates that the developers support the definition of *expectations* and the specification of *boundaries.*

The procedure suggests a systematic method for *gathering the information* from the different units – information that is *correct, necessary* and *accurate.* The procedure divides the educational environment into units for the purpose of gathering information, and uses *communication* techniques to extract whatever information is necessary from the *employees*.

The goal of the elicitation procedure is to *analyse* the current environment so that a different developer could, with this information and his or her understanding of the environment, identify *tasks* and *processes* within the educational domain.

Three of the five phases in the elicitation procedure are concerned with the modelling task. The procedure therefore strongly adheres to the modelling of *business rules*, *workflows* and different *services*. The procedure 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.

Table 3: Characteristics that the procedure 'does not adhere to'

Phase	Characteristic	Rating	Comment
All Phases	Provide **automated support** for the requirements engineering process	NA	While there is no *automated support* developed for the procedure, it should be possible to use existing tools (such as existing Case Tools) to support the documentation process.
	Select appropriate technique for the **problem domain**	PA	The procedure suggested only one way of gathering information. Other techniques such as questionnaires should also be appropriate for the application domain.
	Use of **use cases** to describe related **tasks**	NA	A few resources mentioned this as being important. The procedure did not include use cases to describe scenarios. Object-oriented notation supports the use of use cases.
	Procedures for **maintaining** work products	PA	While the procedure did not specifically mention the importance of maintenance, they support the use of documentation that is easily maintainable.
	Conflict negotiation	NA	No conflict negotiation is mentioned by the procedure.
Elicitation	**Requirement completeness**	PA	Although the procedure does not specifically define measurements to measure requirements completeness, they do suggest a cyclic system that tries to obtain complete requirements.
Modelling	Reflect the **needs of customers / users**	PA	Because the goal of the procedure is to model the current business processes, no need analysis is involved.
	Systems **architecture modelling**	NA	No system architecture modelling is included. This is important during the re-design of current workflows.

There are only a small number of characteristics that the procedure does 'not adhere to'. Table 3 includes all the characteristics that the procedure 'does not adhere to' or 'partially adheres to', with a comment in the last column on each of the ratings.

One characteristic that needs further investigation is the automated support for the requirements engineering process. As we mentioned above, it should be possible to use existing tools, such as existing case tools, to support the documentation process. Furthermore, although the procedure did not specifically mention the importance of maintenance, it supports the use of documentation that is easily maintainable. Cloete et al. (2003) do not give any guidelines on conflict negotiation although this is an important characteristic and research into this is necessary – especially in the educational domain where a diverse group of people is involved in development. The remainder of the characteristics are self-explanatory.

Validation of Requirements Elicitation Procedure

The objective of this work is to identify the characteristics that are required to render functional outputs for an elicitation procedure that could enable successful e-learning implementations in higher education institutions. In Section 3, we described the elicitation procedure for the application domain that was suggested by Cloete *et al.* (2003). Because of the scarcity of published research in this domain, we used this procedure as a basis for our work. However, scientific validation (and possibly augmentation) of the procedure is still necessary if it is to be rendered suitable and valid as an instrument that can be used by other researchers and practitioners. Such an instrument should be able to produce repeatable, usable and effective outputs that could overcome those obstacles to requirements elicitation that contribute to inadequate e-learning implementations.

In attempting to perform such a validation, we conducted a literature study over a wide set of application domains where requirements elicitation is conducted. We demonstrated earlier how requirements elicitation and the subsequent modelling procedure are closely related. The literature review presented us with a list of desirable characteristics that the requirements elicitation and modelling phases in general should possess. By focussing on the specific application domain of this paper, we also extracted a similar, associated list of characteristics. Table 4 summarises these, lists the different phases, and lists the characteristics that the procedure adheres to in the specific phase.

Table 4: Phases in the requirements elicitation procedure, which adhere to the identified characteristics

	Characteristic	Phase 1	Phase 2	Phase 3	Phase 4	Phase 5
All phases	Provide standardised ways of describing work products			√	√	√
	The precision of definition of its notation			√	√	√
	Process model standards			√	√	√
	Support a systematic step-by-step approach	√	√	√	√	√
	Modifiable solutions and iterative in nature	√	√	√	√	√
	Support documentation of requirements	√	√	√	√	√
Elicitation	Requirement relevance		√	√		
	Expectations during specification of requirements		√			
	Correctness		√	√		
	Communication during specification of requirements		√	√		
	Requirement accuracy		√	√		
	Importance of necessity: requirements document	√	√	√		
	Level of control over specifying requirements		√	√		
	Specify constraints / boundaries	√	√	√		
	Support analysis	√	√	√		
	Degree of understanding of the task and process	√	√	√	√	√
	Support data gathering techniques	√	√	√		
	Support customer / client involvement	√	√	√	√	√

Modelling	Motivation for modelling support	√	√	√	√	√
	Model the purpose by describing behaviour				√	√
	Reflect the needs of customers / users	√	√	√	√	√
	Model business rules				√	√
	Support modelling of workflows				√	√
	Clarity of business process				√	√

From Table 4 it is clear that all the phases in the procedure support a systematic approach. It is iterative in nature (the procedure is cyclical and is only completed after a number of iterations). In all the 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. Furthermore, in Phases 3 to 5, a notation used by modellers in process modelling environments, is prescribed. The characteristic *provides standardised ways of describing work products* is therefore adhered to. Similarly, the notation is precise and process model standards are used.

The only characteristic supported in only one phase of the procedure is the *expectations during specification of requirements*. This is understandable because this characteristic is only applicable to the specific phase of the procedure.

Conclusion

The main result of this research is a subjective instrument with fifty-eight indicators aimed at the higher education domain. We attempted to retrieve the indicators or characteristics from authors that commented on the characteristics of requirements elicitation and modelling procedures. We also extracted some from domains such as elicitation or modelling within software engineering or within business process re-engineering. This is, as far as we know, the first research effort that has resulted in an instrument of this nature.

The potential applications of our research results can be discussed from both research and practice perspectives. Researchers may use the instrument as a guideline during the development of similar requirements elicitation procedures. Practitioners using procedures that adprocess re-engineering. This is, as far as we know, the first research effort that has resulted in an instrument of this nature.

The potential applications of our research results can be discussed from both research and practice perspectives. Researchers may use the instrument as a guideline during the development of similar requirements elicitation procedures. Practitioners using procedures that adhere to a set of clearly defined characteristics can do so with the knowledge that the procedure is well-defined, and that it adheres to standards that are used in different application domains.

In further work, we plan to use the instrument to see how other requirements elicitation procedures within the educational domain adhere to the suggested indicators. According to various sources (Finkelstein, Ryan et al. 1996; Maiden and Ncube 1998) we shall in future see the development of reference models for specifying requirements. If this is so, the effort involved in developing requirements models such as ours from scratch will be reduced. This will help move many projects from being creative design to being typical design, and will facilitate the selection of commercial off-the-shelf (COTS) software. Further research into this domain is also necessary in education studies.

Acknowledgements

We wish to thank Dr Elsabé Cloete for her valuable contribution to earlier research leading to this paper. Also thank you to Prof. Remzi Seker who made valuable comments during the development of the requirements elicitation procedure. The work reported on in this article is based partly on work sponsored by a grant from the National Research Foundation of South Africa under Grant Number GUN: 2053850.

Appendix A

Table A1: Activities in a requirements ELICITATION procedure

Reference Number	Reference	Reference Number	Reference
1	(Macaulay 1996)	2	(Madhavji, Holtje et al. 1994)
3	(Goodrich and Olfman 1990)	4	(Eman and Madhavji 1995)
5	(Dawson 1991)	6	(Cordes and Carver 1989)

7	(Davis 1993)	8	(Zagorsky 1990)
9	(Basili and Weiss 1981)	10	(Farbey 1990)
11	(Kotonya and Sommerville 1995)	12	(Nuseibeh and Easterbrook 2000)
13	(Maiden and Rugg 1996)	14	(Johnson 1992)
15	(Schneider and Winters 1998)	16	(Jarke and Kurki-Suonio 1998)
17	(Macaulay, Fowler et al. 1990)	18	(Nuseibeh and Robertson 1997)
19	(Sommerville and Sawyer 1997)	20	(Viller and Sommerville 1999)
21	(Loucopoulos and Kavakli 1995)	22	(Yu 1997)
23	(Greenspan and Feblowitz 1993)	24	(Dardenne, Lamsweerde et al. 1993)
25	(Lamsweerde 2000)	26	(Young 2002)

Table A2: Characteristics

Phase	Sub-phase	Characteristic	No of Refs	References
All phases	Automated support	Provide **automated support** for the requirements engineering process	8	1 2 8 10 11 17 19 26
	Standards	Provide **standardised ways** of describing work products	6	1 2 11 19 26
		The **precision of definition** of its **notation**	2	11 19
		Process model standards	7	4 6 7 12 19 20 25
	Appropriate techniques	**Select appropriate technique** for the **problem domain**	6	2 12 13 19 26
		Use of **use cases** to describe related **tasks**	4	15 16 19 26
		Support a **systematic step-by-step approach**	3	1 19 26
		Solutions can easily be modified and are iterative	3	2 17 26
	Documentation	Support **documentation of requirements**	4	1 10 19 26
	Maintenance	Provide procedures for **maintaining** work products	1	1
	Conflict	**Conflict negotiation**	1	19
Feasibility	Goal Description	Define the **goal** of the modelling	4	3 22 24 25
	Management involvement	**Management consent** with solution	2	2 11
		Management attitude towards **change**	2	9 19
	Feasibility	Support **feasibility studies**	6	1 2 4 11 17 19
		Predictions about the system	1	3
		Scope for integration with existing systems	1	11
		Scope for **evolution**	1	11
	Cost-benefit	Do **cost-benefit analysis** of options	8	1 4 5 6 7 8 10 17
Elicitation	Requirements Specification	**Requirement completeness**	5	3 4 18 19 26

		Requirement relevance	4	2 3 18 26
		Expectations during specification of requirements	4	3 4 21 26
		Correctness	4	6 11 17 18
		Communication during specification of requirements	3	3 11 26
		Requirement accuracy	2	3 26
		Importance of necessity: requirements document	2	6 26
		Level of control over specifying requirements	1	3
	Constraints / Boundaries	Specify constraints / boundaries	5	2 11 12 19 26
	Problem analysis	Support analysis	7	1 2 11 12 19 25 26
		Degree of understanding of the task and process	3	2 3 14
	Use data gathering techniques	Support data gathering techniques	4	2 12 19 26
	Client involvement	Support customer/client involvement	2	3 26
Modelling	Motivation for modelling	Support modelling	11	1 2 3 7 11 12 17 19 20 25 26
	Goal modelling	Model the purpose by describing behaviour	2	21 26
	User involvement	Reflect the needs of customers / users	5	2 4 11 17 26
	Model environment	Model business rules	3	23 25 26
		Support modelling of workflows	3	2 23 25
		Clarity of business process	2	4 18
		Model system services	2	23 25
		Systems architecture modelling	1	19
Triage		Support articulation / coherence of the product concept	3	1 12 18
Verification	Id of Measurement tools	Provide ways of assessing the quality of work products	6	1 2 9 17 18 26
		Enable identification of measures of the requirements engineering process	4	1 2 18 26
		Support descriptions of product effectiveness	2	1 4
	Measures	Quality of the product	1	4
		Process effectiveness	1	5
		Cycle time	1	5
		Trace-ability	1	18
		User/customer satisfaction	5	2 4 9 11 17
		Requirements maturity (number of changes made	3	4 9 10

Bibliography

Basili, V & D Weiss 1981. Evaluation of a Software Requirements Document by analysis of Change Data. *SEKE93: Fifth International Conference on Software Engineering*. San Fransisco Bay, USA.

Bates, AW 2000. *Managing Technological Change, Strategies for College and University Leaders*. San Francisco: Jossey-Bass Publishers.

Bruno, K, B Vrana & L Welz 1998. *Practical Process Engineering for Higher Education. CAUSE98*. Seattle, Washington: CAUSE.

Cloete, E, A Van der Merwe & L Pretorius 2003. A Process Modelling Approach to Requirements Elicitation to Incorporate e-Learning as a Core Learning Strategy. *Seventh World Conference on Integrated Design and Process Technology (IDPT 2003)*. Austin, Texas.

Cordes, D & D Carver 1989. Evaluation Method for User Requirements Documents. *Information and Software Technology* 31(4): 181-188.

Curtis, B, MI Kellner & J Over 1992. Process Modeling. *Communication of the ACM* 35(9): 75-90.

Dardenne, A, A Lamsweerde & S Fickas 1993. Goal-Directed Requirements Acquisition. *Science of Computer Programming* 20: 3-50.

Davis, A 1993. *Software Requirements: Objects, Functions and States*. Englewood Cliffs, NJ: Prentice Hall.

Dawson, J 1991. Toronto Laboratory Requirements Process Reference Guide. Technical Report (Unpublished). Toronto, Canada, IBM Canada Ltd. Laboratory.

Educause 2003. The 2003 *National Survey of Information Technology in US Higher Education*. Washington, DC: Educause.

Eman, KE & NH Madhavji 1995. Requirements Engineering Processes. *International Symposium on Requirements Engineering*. York, England.

Eriksson, H-E & M Penker 2000. *Business Modeling with UML*. New York. John Wiley & Sons.

Farbey, B 1990. Software Quality Metrics: Considerations about Requirements and Requirements Specifications. *Information and Software Technology* 21(1): 60-64.

Finkelstein, A, M Ryan & G Spanoudakis 1996. Software Package Requirements and Procurement. *8th International Workshop on Software Specification and Design*. Schloss Velen, Germany.

Goodrich, V & L Olfman 1990. An Experimental Evaluation of Task and Methodology Variables for Requirements Definition Phase Success. *Proceedings of the Twenty-Third Annual Hawaii International Conference on System Sciences*. Hawaii. IEEE Computer Society.

Greenspan, S & M Feblowitz 1993. Requirements Engineering Using the SOS Paradigm. *1st International SYmposium on Requirements Engineering (RE'93)*. San Diego, USA, 4-6 January 1993.

Hickey, AM & AM Davis 2003. Requirements Elicitations and Elicitation Technique Selection: A Model for Two Knowledge-Intensive Software Development Processes. 36th Hawaii International Conference on System Sciences. Hawaii. IEEE.

IEEE 1998. IEEE Guide for Developing System Requirements Specifications. New York, USA.

Jarke, M & R Kurki-Suonio 1998. Guest Editorial - Special Issue Scenario Management. *IEEE Transactions on Software Engineering* 24(12).

Johnson, P, 1992. Human-Computer Interaction: psychology, Task Analysis and Software Engineering. Maidenhead: McGraw-Hill.

Kotonya, G & I Sommerville 1995. *Requirements Engineering with Viewpoints*. Lancaster, UK: Cooperative Systems Engineering Group: CSEG/10/1995.

Lamsweerde, AV 2000. Requirements Engineering in the Year 00: A Research Perspective. *International Conference on Software Engineering*. Lumerick, Ireland.

Laurillard, D 1993. *Rethinking University Teaching*. London: Routledge.

Loucopoulos, P & E Kavakli 1995. Enterprise Modelling and the Teleological Approach to Requirements Engineering. *International Journal of Intelligent and Cooperative Information Systems* 4(1): 45-79.

Luker, MA 2000. P*reparing Your Campus for a Networked Future*. San Fransisco. Jossey-Bass Inc.

Macaulay, L 1996. *Requirements Engineering*. Great Britain: Springer-Verlag.

Macaulay, L, C Fowler, M Kirby & A Hutt 1990. USTM: A New Approach to Requirements Specification. *Interacting with Computers* 2(1): 92-118.

Madhavji, N, D Holtje, W Hong & T Bruckhaus 1994. Elicit: A Method for Eliciting Process Models. *3rd International Conference on Software Process*. Reston, Virginia.

Maiden, N & G Rugg 1996. ACRE: Selecting Methods for Requirements Acquisition. *Software Engineering Journal* 11(3): 183-192.

Maiden, N A M & C Ncube 1998. Acquiring Requirements for Commercial Off-The-Shelf Package Selection. *IEEE Software* 15(2): 45-56.

McDermid, J 1993. Software Engineer's Reference Book. Oxford. Butterworth-Heinemann.

Nuseibeh, B & S Easterbrook, 2000. Requirements Engineering: A Roadmap. *International Conference on Software Engineering*. Lumerick, Ireland.

Nuseibeh, B & S Robertson 1997. Making Requirements Measurable. *International Conference on Software Engineering* (ICSE 1997). Boston, MA USA.

Pressman, RS 2000. *Software Engineering, a Practitioner's Approach.* United Kingdom: McGraw-Hill International.

Ryan, S, B Scott, H Freeman & D Patel 2000. *The Virtual University - The Internet and Resource-Based Learning.* London: Kogan Page.

Rzepka, WE 1989. A Requirements Engineering Testbed: Concept Status, and First Results. *Twenty-Second Annual Hawaii International Conference on System Sciences.* Hawaii. IEEE Computer Society.

Schneider, G & J Winters, 1998. Applying Use Cases: a practical guide. Boston, MA: Addison-Wesley.

Sommerville, I & P Sawyer 1997. *Requirements Engineering: A Good Practice Guide.* UK: John Wiley & Sons.

Tait, F 1999. Enterprise Process Engineering: A template tailored for higher education. *CAUSE/EFFECT* 22(1).

Viller, S & I Sommerville 1999. Social Analysis in the Requirements Engineering Process: from Ethnography to Method. *4th International Symposium on Requirements Engineering.* Limerick, Ireland, 7-11th June 1999.

Whitten, JL, L D Bentley & K C Dittman 2001. *System Analysis and Design Methods.* New York: McGraw-Hill Higher Education.

Young, R, 2002. Recommended Requirements Gathering Practices. *Crosstalk: The Journal of Defense Software Engineering* April 2002: 9-12.

Yu, E, 1997. Towards Modelling and Reasoning Support for Early-Phase Requirements Engineering. *3rd IEEE International Symposium on Requirements Engineering (RE'97).* Annapolis, USA, 6-10 January 1997.

Zagorsky, C, 1990. Case Study: Managing the Change to CASE. *Journal of Information Systems Management.* September 1990: 24-32.

Authors' Contact Details

Alta van der Merwe (vdmeraj@unisa.ac.za)
Paula Kotzé (kotzep@unisa.ac.za)
School of Computing, University of South Africa
Johannes Cronjé (jcronje@up.ac.za)
Department of Teaching and Training Studies, University of Pretoria, Pretoria, South Africa