Authors	Alta van der Merwe, Johannes Cronje Paula Kotze
Paper Name	The Functionality of a Requirements Elicitation Procedure Developed for Process Modelling within the Higher Education Application Domain
Year	2004
Type	Conference Proceedings
Conference	South African Computer Lecturer Assciation
Editors	R Klopper M Haharaj
Place	Durban, South Africa

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

Alta van der Merwe[1], Johannes Cronje[2], Paula Kotzé[1]
[1]School of Computing, University of South Africa, {vdmeraj,kotzep}@unisa.ac.za
[2]Department of Teaching and Training Studies, University of Pretoria, jcronje@up.ac.za

Abstract

Although different application domains use requirements elicitation procedures to gather information and model the application domain, few guidelines exist with regard to the important characteristics that these procedures should adhere too. This paper focuses on the identification of a set of characteristics for a functional requirements elicitation procedure within the higher education domain.

1 Introduction

The Internet as innovation is no longer an inceptive technology in the higher education institutions application domain. In a recent report published by Educause (Educause, 2003), an increase on the number of institutions in the USA that uses the Internet to provide web-based campus portals was reported to rise 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 noticed in South Africa.

Traditional higher education institutions that have already incorporated e-learning into their curricula often claim to have a competitive advantage by serving a much wider audience of students. To stay competitive institutions should actively be involved in actions to convert current processes to provide similar or better electronic services to its client base, than that of its competitors (Bates, 2000; Laurillard, 1993). However, the incorporation of e-learning into the curricula of traditional higher education institutions is not a trivial task (Luker, 2000; RyanScottFreeman *et al.*, 2000). The main cause of the dissatisfaction with e-learning, and sometimes inefficient implementations, is similar to those in most application domains where developments are not successfully implemented (Pressman, 2000; WhittenBentley and Dittman, 2001). Reluctance to incorporate electronic innovations often originates in the failure of completely understanding the application domain, which requires careful consideration of e-learning technologies and strategies, if it is to be successful.

Requirements elicitation is a technique used by different organizations to describe and specify an application domain. Various requirements elicitation procedures exist to gather information and model environments in different application domains. For example, in software development projects there is a number of software requirements engineering procedures (Hickey and Davis, 2003; McDermid, 1993; Pressman, 2000), and in business process re-engineering authors such as Davenport (1993) and Hammer (1990) describe specialized re-engineering elicitation procedures. Furthermore, in software and business environments, numerous guidelines exist which describe the characteristics of the procedures. Although it therefore seems that requirements elicitation procedures would be the ideal tool to use to try to build an understanding of the higher educational domain, only a very limited number of descriptions of process modelling procedures exist with regard to modelling of this environment, or of the characteristics of a requirements elicitation procedure for this domain (BrunoVrana and Welz, 1998; CloeteVan der Merwe and Pretorius, 2003; Tait, 1999).

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*, as it will assist them to establish a procedure that includes all the important traits required from such procedure. The recognition and inclusion of these important traits will lead to the development of improved products, such as re-engineering of the current environment to include e-learning technologies, hence improving the chances of successful deployment and acceptance of these products. The second constituent that may benefit from this paper is *researchers*, who not only stand at the base of establishing and developing new knowledge for the interest of society, but also join hands with practitioners to define well-needed standards.

Section 2 of this paper identifies the context of the paper with regards 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. Section 3 describes a requirements elicitation procedure as applied to higher education environment, while Section 4 aims to show how this procedure adheres to the suggested characteristics. Section 5 addresses the issue of scientific validation of the reported research, while Section 6 concludes.

2 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). For a requirements elicitation procedure to be considered effective, it should at least produce the initial goal (Rzepka, 1989). *Process modelling* presents a technique (involving several activities) to graphically depict the series of processes that accomplish a predefined goal (CurtisKellner and Over, 1992; Snowdown, Accessed 16 April 2002). The *process model* is the *structure* that represents a group of processes and their relationship to one another, which together accomplishes a specific goal. These two sub-fields naturally exist within cyclic methodologies that have the aim to develop software or to re-engineer current environments (Hickey and Davis, 2003; Pressman, 2000). Our focus is on the elicitation and modelling activities, as illustrated in Figure 1.

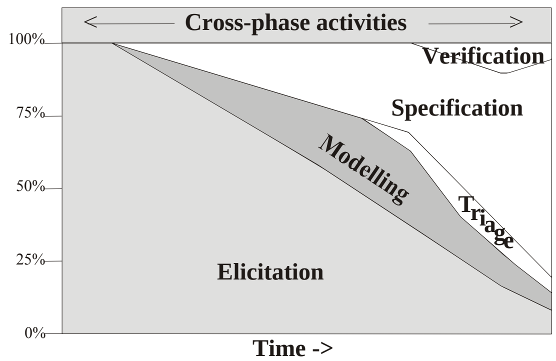

Figure 1: Requirements elicitation activities procedure (adapted from (Hickey and Davis, 2003))

When identifying the required characteristics of an elicitation procedure, we conducted a comprehensive literature review of sources referring to characteristics within the field of study and which comment on the characteristics of a range of aspects of 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 characteristics identified. After these cycles, the characteristics became repetitive to such a nature that we did not add any new characteristics to the list.

Table A1 (Appendix A) gives a list of the twenty-six resources, with a number assigned to each for further referral. Table A2 (Appendix A) specifies the list of characteristics 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 include (1) a feasibility study, (2) elicitation, (3) modelling, (4) triage, (5) verification and (6) cross-phase activities (Hickey and Davis, 2003; Macaulay, 1996; Sommerville and Sawyer, 1997). We found that some of the characteristics identified, even if mentioned initially as being important for requirements elicitation, actually belonged to other activities such as a feasibility stage, and not really an elicitation stage. We used these core steps and categorized each of the characteristics into one of these steps to ensure that we only list characteristics that belong to the focus of our study, namely the requirements elicitation and modelling phase (Appendix A, Table A2). We also merged characteristics with the same meaning, to end 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).

Table 1: List of characteristics

Phase	Sub-phase	Characteristic
All Phases	Support	Provide **automated support** for the RE 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 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**

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 is limited to the phases mentioned.

3 A Requirements Elicitation Procedure for the Higher Education Environment

We now present an overview of a requirements elicitation procedure with the goal to model a higher education environment. This procedure was developed and tested as part of a research project 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 (Cloete *et al.*, 2003) – the result of Phase 1 is the establishment of objectives, whereas the identification of critical institutional units (Phase 2) and the identification of primary processes (Phase 3) assist in understanding the domain. It is also during these last two phases, that the developers collect stakeholder requirements. 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). We subsequently describe each of these phases in more detail.

3.1 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 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 goal(s) as well as a clear specification of the required deliverables.

3.2 Phase 2: Identifying the Critical Institutional Units

The objective of the procedure is to identify the critical processes in the application domain in order to examine their essential activities and workflow, so that the application domain can be understood. The critical processes can only be identified by considering different operational units within the institution. A unit refers to a working segment of the institution that is responsible for specific tasks, for example, a financial section, an academic department, a technical division, et cetera. As a first step, all these units within the institution are listed. This list can be compiled by studying documentation and diagrams, such as organizational diagrams, or through interviews. With the focus on possible adoption and integration of e-learning, the second step accomplishes extracting those units that are actively involved in the creation and presentation of learning environments. The units focusing on other aspects of the institution are then labelled as support units and deleted from the unit list. For example, the catering services department prepares food but it is not directly responsible for, or involved in the learning environment, therefore, it will be removed form the unit list. The deliverable of Phase 2 is a list of the critical operational units of an institution.

3.3 Phase 3: Identify Primary Processes

In the next three phases, the procedure uses 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 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 suggested a starting list of primary processes for the e-learning domain as being 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 are suggested to expand the list and to 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 is on the goals to be achieved rather than on the individual activities that 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, 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. Categorize each process as either being a support or a primary process using to the definitions provided earlier.
3. Attempt a mapping for each of the newly identified primary processes to an item on the starting list. 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 set of the identified primary processes, namely:

$$\{P_j\}_{j=1}^{m} \text{ for } j, m \in N_1$$

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 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.

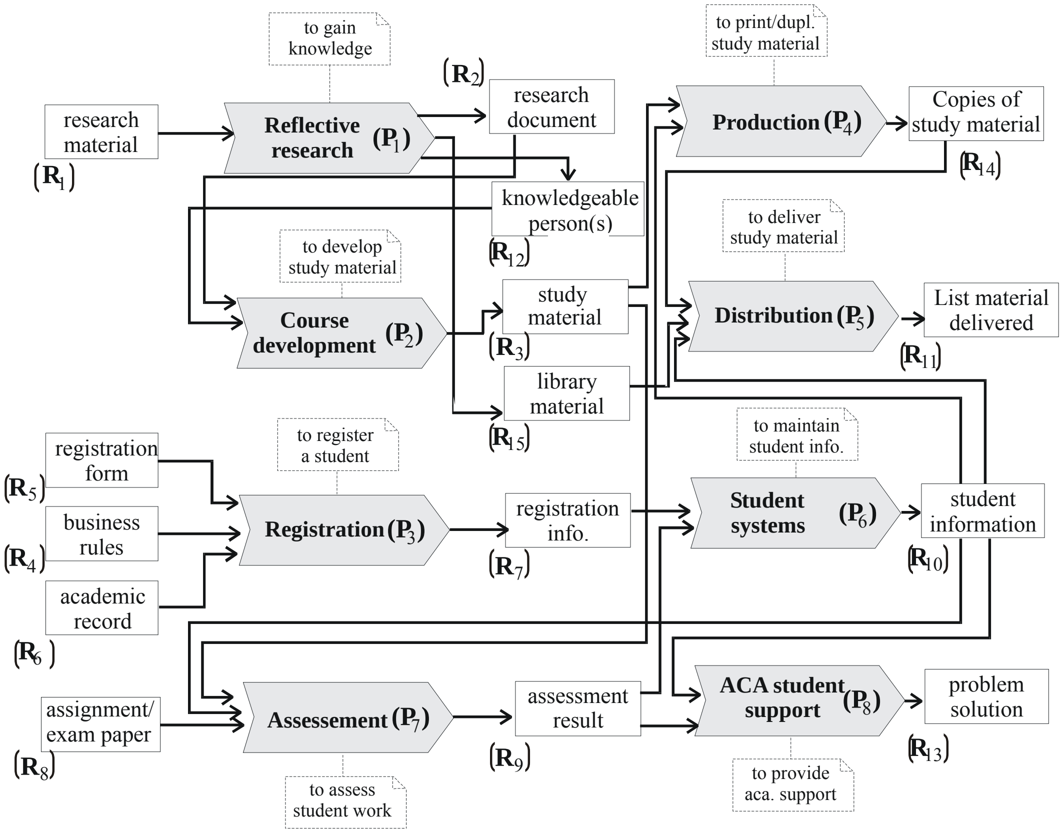

Figure 3: High-level process model for the higher education domain

3.4 *Phase 4: Construct the High-level Process Model*

For constructing the high-level process model, the procedure used standard notation that include the process itself, process resources and the goal description of the process (Eriksson and Penker, 2000). 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 potentially serve as an input resource to another process. Each process has at least one input resource and one output resource associated with it. The first step toward constructing the high-level process model is to define the goal, input resources, and output resources associated with each item on the process list, which was 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:

$$\{R_k\}_{k=1}^{n} \text{ for } k, n \in N_1$$

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 as long as there is only a small number of primary processes to consider and can be accomplished simply by connecting related processes through directed lines. However, as the number of primary processes increases the complexity to depict the workflow also increases. In such a case, the procedure suggests a more formal approach to establish relationships between primary processes.

The objective is to identify the resources that serve as both input and output resource for the different processes and then to 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_{jk}) that a resource R_k has with a process P_j (for all *j* and all *k*). These association values may be INPUT (T_{jk} = I), OUTPUT *(*T_{jk} = O), or NO ASSOCIATION (T_{jk} = NA). Each T_{jk} is then stored as a entry in a process-resource table, which vertically tabulates all processes from top to bottom and horizontally tabulates all resources from left to right.

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

- For $j = 1..m$ and $k = 1..n$, describe the all the resources in terms of their association values with P_j. This is written as a triplet (P_j, R_k, T_{jk}).
- For $j = 1..m$, graphically depict P_j on a process model diagram with its associated goal.
- For $k = 1..n$, add the identified resources R_k to the diagram.
- Use the set of triplets (P_j, R_k, T_{jk}), 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 illustrated in Figure 3.

3.5 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 into further sub-processes.

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

- 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.
- Associate the sub-processes with one another through input and output resources as described in Phase 4.
- Draw the process model, which graphically depicts the sub-processes and their relationships between one another.

Repeat these steps for each of the identified sub-processes until all processes are atomic or 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.

4 How does the Requirements Elicitation Procedure Adhere to the Characteristics Identified?

In this section, we consider the requirements elicitation procedure described in Section 3 and describe its scientific soundness against the characteristics described in Section 2. Using the characteristics identified in Table 1, each phase of the requirements elicitation procedure is evaluated as either strongly correlating with the characteristic, partially correlating, or no correlation what so ever. With the characteristics identified in Table 1 as starting point, we rated the requirement elicitation procedure with relation to each characteristic, as:

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.

The result of this rating of the different aspects of the requirement 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 rating achieved for each characteristic.

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 to be *iterative*. More than once in the procedure it refers to the output of a phase 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, modelling the primary processes that are important in creating a learning environment. This goal and the limitations are expected in the beginning of the procedure, which indicates that the developers support the definition of *expectations* and the specification of *boundaries.*

The procedure suggested a systematic procedure to *gather the necessary information* from the different units, with the goal to gather *correct, necessary* and *accurate* information. The procedure 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 the elicitation procedure 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 the elicitation procedure are concerned with the modelling task. The procedure therefore strongly adheres to the modelling of *business rules*, the *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.

There is 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 adhere to', with a comment in the last column on each of the ratings.

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

	Sub-phase	**Characteristic**	Not Adhere (NA)	Partially Adhere (PA)	Strongly Adhere (SA)
All Phases	Support	Provide **automated support** for the RE 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 nature**			√
	Documentation	Support **documentation of requirements**			√
	Maintenance	Procedures for **maintaining** work products		√	
	Conflict	**Conflict negotiation**	√		
	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			√
Elicitation	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**	√		

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

Phase	Characteristic	Rating	Comment
All Phases	Provide **automated support** for the RE process.	NA	There is no *automated support* developed for the procedure, although it should be possible to use existing tools to support the documentation process, such as existing Case Tools.
	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	The procedure did not specifically mention the importance of maintenance, but they support the use of documentation, which is easily maintainable.
	Conflict negotiation.	NA	**No conflict negotiation is mentioned by the procedure.**
Elicitation	**Requirement completeness**	PA	**The procedure do not specifically define measurements to measure requirements completeness, they do suggest a cyclic system, which try to get complete requirements.**
Modelling	Reflect the **needs of customers / users.**	PA	Because the goal of the procedure is to model the current business processes, there is no need analysis involved.
	Systems **architecture modelling.**	NA	No system architecture modelling included. Important during re-design of current workflows.

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. Furthermore, the procedure did not specifically mention the importance of maintenance, but support the use of documentation, which is easily maintainable. Cloete et al. (2003) do not give any guidelines on conflict negotiation, which are an important characteristic, and research into this is necessary especially in the educational domain with diverse personnel involved in development. The rest of the characteristics are self-explanatory.

5 Validation of Requirements Elicitation Procedure

The objective of this work was to identify the required characteristics to render functional outputs for an elicitation procedure that could enable successful e-learning implementations in higher education institutions. In Section 3, we have described the elicitation procedure for the application domain that was suggested by Cloete et.al (2003). Due to the scarceness of published research in this domain, we used this procedure as a basis of our work. However, scientific validation (and possibly augmentation) of the procedure is still necessary in order to render it 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 the obstacles in requirements elicitation that contribute to insufficient e-learning implementations.

In attempt to perform such a validation, we performed a literature study over a wide set of application domains where requirement elicitation is conducted. We showed earlier how requirement elicitation and the subsequent modelling procedure go hand in hand. From the review, we came up with a list of desirable characteristics that the requirement elicitation and modelling phases in general should possess. Focussing on the specific application domain of this paper, we also extracted a similar associated list of characteristics. Table 4 gives a summary of these with the different phases listed and the characteristics listed that the procedure adheres to in the specific phase.

From the table it is clear that all the phases in the procedure support a systematic approach. It is iterative in nature (the procedure is cyclic and only completed after a number of iterations). In all the phases the information gathered by the developers are documented, which indicates that the procedure do support 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 are prescribed. Therefore, the characteristic *provides standardised ways of describing work products* is 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 the characteristic is only applicable to the specific phase of the procedure.

Table 4: Phases in the requirements elicitation procedure which adheres 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 be **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 to **Support modelling**	√	√	√	√	√
	Model the **purpose** by describing **behaviour**				√	√
	Reflect the **needs of customers / users**	√	√	√	√	√
	Model **business rules**				√	√
	Support **modelling of workflows**				√	√
	Clarity of business process				√	√

6 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. However, we 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 resulted in an instrument of this nature.

The potential applications of our research results can be discussed from both the 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, adhering to standards 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 various sources (FinkelsteinRyan and Spanoudakis, 1996; Maiden and Ncube, 1998) the future see the development of reference models for specifying requirements, so that the effort of developing requirements models, such as ours, from scratch, will be reduced. This will help move many projects from being creative design to being normal design, and will facilitate the selection of commercial off-the-shelf (COTS) software. Further research in this problem domain is also necessary concerning the educational domain.

Acknowledgements

We wish to thank Dr. Elsabé Cloete, former study leader of Alta van der Merwe, who left the University of South Africa at the beginning of 2004, for her valuable contribution not only to this paper, but also to previous research. The work reported upon in this article is partly based on work sponsored by a grant from the National Research Foundation of South Africa under Grant Number GUN: 2053850.

Appendix A

Reference Number	Reference	Reference Number	Reference
1	(Macaulay, 1996)	2	(MadhavjiHoltjeHong *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	(MacaulayFowlerKirby *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	(DardenneLamsweerde and Fickas, 1993)
25	(Lamsweerde, 2000)	26	(Young, 2002)

Table A1: Activities in a requirements ELICITATION procedure

Phase	Sub-phase	Characteristic	No of Refs	References
All phases	Automated support	Provide **automated support** for the RE 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
		Modifiable design solutions and be **iterative in nature**	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 RE process	4	1 2 18 26
		Support descriptions of **product effectiveness** in RE terms	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 to R document)	3	4 9 10

Table A2: Characteristics

Bibliography

Basili, V. and 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, A. W. (2000) *Managing Technological Change, Stragegies for College and University leaders.* San Francisco: Jossey-Bass Publishers.

Bruno, K., B. Vrana and L. Welz (1998). Practical Process Engineering for Higher Education. CAUSE98, Seattle, Washington, CAUSE.

Cloete, E., A. Van der Merwe and 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. and D. Carver. (1989) Evaluation method for user requirements documents. *Information and Software Technology*, **31**(4): 181-188.
Curtis, B., M. I. Kellner and J. Over. (1992) Process Modeling. *Communication of the ACM*, **35**(9): 75-90.
Dardenne, A., A. Lamsweerde and 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, N,J: 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, K. E. and N. H. Madhavji (1995). Requirements Engineering Processes. International Symposium on Requirements Engineering, York, England.
Eriksson, H.-E. and 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 and G. Spanoudakis (1996). Software Package Requirements and Procurement. 8th International Workshop on Software Specification and design, Schloss Velen, Germany.
Goodrich, V. and 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. and 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, A. M. and A. M. 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. and 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. and I. Sommerville. 1995. Requirements Engineering With Viewpoints. Lancaster, UK, Cooperative Systems Engineering Group: CSEG/10/1995.
Lamsweerde, A. v. (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. and 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, M. A. (2000) *Preparing your campus for a networked future.* San Fransisco: Jossey-Bass Inc.
Macaulay, L. (1996) *Requirements Engineering.* Great Britian: Springer-Verlag.
Macaulay, L., C. Fowler, M. Kirby and A. Hutt. (1990) USTM: A New Approach to Requirements Specification. *Interacting with Computers*, **2**(1): 92-118.
Madhavji, N., D. Holtje, W. Hong and T. Bruckhaus (1994). Elicit: A method for Eliciting Process Models. 3rd International Conference on Software Process, Reston, Virginia.
Maiden, N. and G. Rugg. (1996) ACRE: Selecting Methods For Requirements Acquisition. *Software Engineering Journal*, **11**(3): 183-192.
Maiden, N. A. M. and 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. and S. Easterbrook (2000). Requirements Engineering: A Roadmap. International Conference on Software Engineering, Lumerick, Ireland.
Nuseibeh, B. and S. Robertson (1997). Making Requirements Measurable. International Conference on Software Engineering (ICSE 1997), Boston, MA USA.
Pressman, R. S. (2000) *Software Engineering, A Practitioner's Approach.* Fifth. United Kingdom: McGraw-Hill International.
Ryan, S., B. Scott, H. Freeman and D. Patel. (2000) *The Virtual University - The Internet and Resource-based learning.* London: Kogan Page.
Rzepka, W. E. (1989). A Requirements Egnineering Testbed: Concept Status, and First Results. Twenty-Second Annual Hawaii International Conference on System Sciences, Hawaii, IEEE Computer Society.
Schneider, G. and J. Winters. (1998) *Applying Use Cases: a practical guide.* Boston, MA: Addison-Wesley.
Snowdown, R. A. (Accessed 16 April 2002) *Overview of Process Modelling*. [Online]. Available from. [Accessed 16 April].
Sommerville, I. and 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. and 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, J. L., L. D. Bentley and K. C. Dittman. (2001) *System Analysis and design methods.* 5th Edition. 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.