Authors	Alta van der Merwe,
Paper Name	A requirements elicitation methodology for educational process modelling
Year	2003
Type	Conference Proceedings
Conference	4th International Conference on Information Communication Technologies in Education
Place	Samos, Greece

A REQUIREMENTS ELICITATION METHODOLOGY FOR EDUCATIONAL PROCESS MODELLING

Alta van der Merwe
Computer Science and Information Systems
University of South Africa
Pretoria, South Africa
Tel: (29) 12 429 6339
E-mail: vdmeraj@unisa.ac.za

Abstract

Approaches used during the development of methodologies differ from analyst to analyst and depend on various factors including the application domain, the availability of existing methodologies and budget constraints. In this paper, the author proposes a requirements elicitation methodology that may be used by educational institutions during the requirements engineering process to gain a complete understanding of the critical processes (and their sub-processes) in the application domain. The purpose of the methodology is to accelerate the requirements engineering process and, as a result, reduce the initial cost of systems development.

Keywords: educational modelling, organizational model, requirements elicitation, e-learning

Introduction

A methodology consists of a prescribed set of tasks or steps undertaken by a development team to achieve a desired result. It focuses on processes and techniques used in a specific discipline, and includes repeatable best practices, templates and prior accumulated knowledge. It also creates a well-ordered environment that allows for a better chance of success in developing a solution for the application domain. An institution may purchase a methodology from a vendor or may prefer to develop a methodology in-house to meet the specific needs of an application domain.

The focus of this paper is on a *requirements elicitation methodology,* which may be used during the requirements engineering process at educational institutions to gain a complete understanding of the *critical processes* (and their sub-processes) in the application domain. Critical processes, in this domain application, are those involved in creating and presenting a learning environment. The requirements elicitation methodology was developed at the University of South Africa (UNISA) and its functionality was tested at other institutions.

In developing a methodology, developers may use different approaches. The following are some of these approaches:

- The documentation of best practices, which is based on previous experiences.
- The proposal of a new methodology, which is implemented and evaluated.
- The adoption of an existing methodology, which is changed to fit the particular needs of the educational institution.
- The recording of methods used to acquire a set of pre-defined deliverables.

Our deliverable was to record a method that successfully delivers a set of process models, through which the user may gain a better understanding of the core processes (and sub-processes) of the institution. The development team[1] decided to use the last approach on the list in developing the methodology.

The paper proceeds by discussing the approach used at the University of South Africa to develop the requirements elicitation methodology and then goes on to discuss the suggested methodology. This is followed by an overview of the implementation of the methodology at other educational institutions. Finally we conclude with a short review and acknowledgement at the end of the paper.

Development approachAs already mentioned, the development team studied the current application domain and from this environment developed the requirements elicitation methodology. Our goal was to gain a complete understanding of the critical processes (and their sub-processes) in the application domain. This understanding was possible through identification of the different core processes, sub-processes and the identification of the workflow between them.

The development process included different activities. The development team first studied existing requirements elicitation methods and techniques used in different requirements engineering methodologies. This activity was followed by the identification of existing formal requirement elicitation methods currently used at Unisa. It was found that a number of different requirement elicitation methods are used at the institution to gather information for specific projects, e.g. for course design and systems development. However, the methods used at the institution focused on different problem domains and none was found with the specific goal of identifying the core processes (and sub-processes) within the institution.

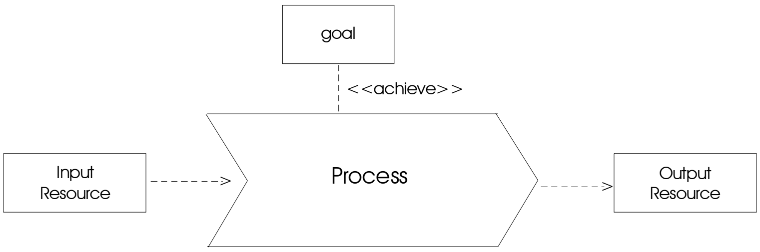
It is appropriate at this stage to comment on the *information gathering* techniques used during the development process. There is a wide variety of information-gathering techniques available, including

[1] More than one person was involved in the development initiative and they are mentioned at the end of this paper. For the purpose of this paper they are referred to as the ‘development team’.

interviews, workshops, prototyping, etc (Kononya and Sommerville, 1998; Maciaszek, 2001; Atlee and Berry, 2002). We used three different information-gathering techniques, namely interviews, the study of existing documentation and direct observation within the application domain.

The requirements elicitation methodology was developed using a bottom-up approach, starting with the deliverable under consideration, namely the processes within the institution. According to Kononya and Summerville (1998), a process is an "organised set of activities, which transforms inputs to outputs". There are a number of significant elements that may be used to depict a particular process, for example the process itself and its resources. Different process modelling methodologies suggest different significant elements to depict a particular process, depending on the application domain. We suggest the following elements from Eriksson and Penker's (Eriksson and Penker 2000) list: of the process, resources and the goal description. The notation used in graphical representations to represent a process, is given in Figure 1.

Figure 1: The process notation used in graphical representations



Before we could model the different processes and the flow between them, we needed a list of them. Subsequently, our first task was to compile a list of all the processes within the institution. After listing all the processes (to our knowledge), we needed a course of action to ensure that the process list included all possible processes within the institution. The institutional structure was consulted and representatives identified within each unit[2] to compare the list with their own list of responsibilities (processes within the unit). Any processes neglected in the first round of process identification were added after this activity.

Our next step was to group processes together to distinguish between core processes and sub-processes. In the grouping process, processes that belong together were categorized together, for

[2] 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.

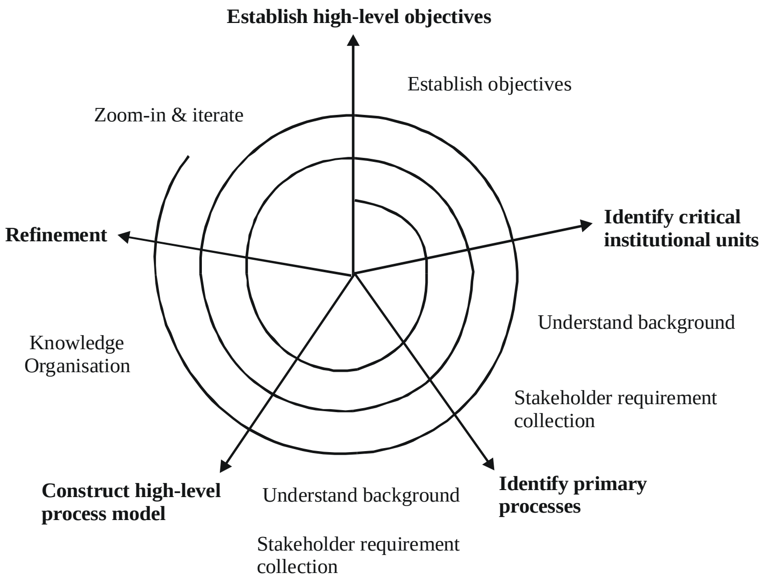
example, the atomic activities[3] *answering student e-mail* and *answering postal queries* were grouped together under a higher-level process, namely *answer student queries.* The identification and categorization of these processes led to the identification of a core list of high-level processes given in the following section.

As mentioned, the purpose of a process is to convert an input into an output. We used these inputs and outputs to indicate the workflow between the different processes. This task remains simple as long as there are only a small number of processes to consider. However, as the number of processes increases, the complexity of depicting the workflow accurately also increases considerably. To cover such a case, we developed a more formal approach to the establishment of relationships between processes. The approach is discussed in more detail in the next section.

The requirements elicitation methodology

Kotonya & Summerville (1998) suggest four critical activities to be included in a good requirements elicitation procedure, namely objective setting, background knowledge acquisition, knowledge organisation and stakeholder requirements recollection. We incorporated these critical activities into our proposed methodology, which consists of five separate phases, illustrated in Figure 2 and described in more detail in the remainder of the document.

Figure 2: Proposed requirements elicitation methodology



(Cloete, Van der Merwe et al., 2003)

Figure 2 shows the phases of our methodology as a spiral model, in which the different phases are interleaved and revisited many times to

---

[3] Any sub-process that could not be broken into further sub-processes was referred to as an *atomic activity.*

build a complete high-level process model. 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) play an important role in the understanding of the application domain. Another activity is the collection of stakeholder requirements during these two phases. The acquired information is organised into a *high-level process model* during Phase 4, followed in Phase 5 by the *refinement of the process model* into a number of sub-processes. We subsequently describe each phase.

Phase 1: Initialization

The reason for building a model of the problem domain (in our case specifically the educational problem domain) is often vaguely defined at the onset of the modelling exercise. In Phase 1, the development team, in co-operation with management, compiles a detailed description of the project.

The deliverable of this phase is a descriptive *document* serving as a framework, which is 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*.

Phase 2: Identify institutional units

In the succeeding phase, Phase 3, we need to identify the processes involved in the different institutional units. To be able to identify the different processes, we first need to identify the relevant units in the institution, which is the purpose of this phase.

The first step in this phase is the listing of all the units in the institution. The second step is to identify (from the list compiled) only the units actively involved in creating and presenting a learning environment. The remaining units that are not actively involved in this activity are categorized as support units and are deleted from the unit list. The deliverable of this phase is a document listing the relevant units (departments and sections) in the institution.

Phase 3: Identify the primary processes

Processes in the educational environment can either be *primary* or *support* processes. Primary processes are those critical activities responsible for the design and construction of the student's learning environment. Support processes provide sustenance for the primary processes; in other words, they play a secondary role in accomplishing the defined goal. The purpose of Phase 3 is to identify the primary processes of the problem domain.

Porter (Porter, 1985) identifies five *primary activities* in the business environment contributing to the value of businesses, namely inbound logistics, production, outbound logistics, marketing & sales, and service. We apply the fundamentals of his work to the educational problem domain and describe a list of primary processes, rather than primary activities, suitable for the educational environment. This list should be considered as a starting list only since it can be expanded or adjusted according to specific needs in the particular problem domain. The elements of this list include (1) the registration process, (2) the development of course material, (3) the production of course material, (4) the distribution of course material, and (5) the academic support available to the student.

The following steps can be used to expand the list and to verify its adequacy and completeness:

1. List and document the most important processes of the particular unit.
2. Categorize each process as being either a support process or a primary process.
3. For each responsibility categorized as primary (step 2 above), map it to the ones given in the starting list. Those that cannot be mapped are added to the list as additional primary processes.

Phase 4: Build the high-level process model

As previously discussed, the high-level process model consists of all the primary processes in the institution. Thus, each of the primary processes (identified in Phase 3) must be included in the process model. The first step in building the high-level process model is to define the goal, input resources and output resources for each of the primary processes.

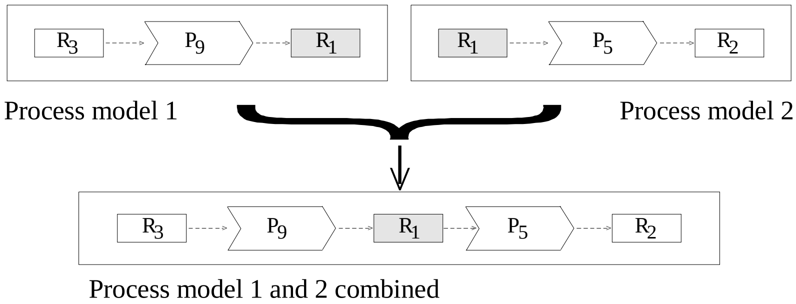
The second step is to link the different primary processes with one another through input and output resources. The task of linking a few (two or three) primary processes is relatively simple. However, for more than five processes this task becomes considerably more complex. In such a case, a specific approach to establishing relationships between primary processes can be helpful. Below we describe our approach to resolving complexities in establishing relationships between primary processes.

We first take cognisance of what we have at this stage:

- A set of primary processes: $\{P_k\}_{k=1}^{m}$ with k,m $\in \mathbb{N}$ (m is the total number of processes)
- A set of resources: $\{R_j\}_{j=1}^{n}$ with j,n $\in \mathbb{N}$ (n is the total number of resources)

Our objective is to identify which resources serve as both input and output resource for different processes and then to eliminate redundant resources (those resources that would appear more than once on the same process model diagram). For example, if R_1 serves as input resource for process P_5 and also as an output resource for process P_9, a process model diagram portraying these two processes will have two occurrences of R_1 on the same diagram. To reduce redundancy as well as complexity, one of the occurrences of R_1 can be removed and a link inserted that connects both processes to the one occurrence of R_1. The direction of the link indicates whether the resource is an input or output of the particular process. Figure 3 illustrates the concept.

Figure 3: Shared resource R_1 links resource P_9 and P_5



We are interested in determining whether each resource R_j is an input resource or an output resource for every primary process. This information (regarding the value of the resource) is stored in a *process-resource table*, enabling the development team to associate different processes with one another through common resources. Table 2 illustrates this idea.

Table 1: Process-resource table

	R_1	R_2	….	R_j	…	R_n
P_1	T_{11}	T_{12}		T_{1j}		T_{1n}
P_2	T_{21}	T_{22}		T_{2j}		T_{2n}
….						
P_k	T_{k1}	T_{k2}		T_{kj}		T_{kn}
…						
P_m	T_{m1}	T_{m2}		T_{mj}		T_{mn}

We define T_{kj} as the resource value (which can be either *INPUT* or *OUTPUT*) indicating how resource R_j is associated with process P_k. For example, in our previous example R_1 is an input resource for process P_5, thus T_{51} = INPUT. In the same way R_1 was also the output of P_9, which means that T_{91} = OUTPUT. Also, R_3 was the input for process P_9, which means T_{93} = INPUT, but T_{53} will have no value because there is no relation between R_3 and P_5. We illustrate this example in Table 2.

Table 2: Process-resource table for example in Figure 3

	R_1	R_2	R_3	….
….				
P_5	INPUT	OUTPUT		
…				
P_9	OUTPUT		INPUT	
…				

To link the processes, we need to repeat the following steps for each resource R_j:

1. For each process P_k, where T_{kj} <> NILL, describe the resources for the process in terms of the association and direction of each resource. This can be written as a triplet (P_k, R_j, T_{kj}).
2. Graphically depict each process P_k (for all k) on the diagram (only one instance of any P_k) with its goal.
3. Add all the identified resources, Rj (for all j) to the diagram (do not add direction between resources and processes yet).
4. Use the set of triplets (identified in step 1) to add direction between processes and resources.

The deliverable of this step is the high-level process model with the primary processes and their relationship modelled on a grammatical diagram.

Phase 5: Refine the process model

As has been mentioned, during Phase 4 we included only the *primary processes* and their relation to one another in the high-level process model. Each of the primary processes in the process model consists of a number of sub-processes used to accomplish the pre-defined *goal* associated with the process. The purpose of this phase is to decompose and particularise the processes in the high-level process model through iterative steps into a set of sub-processes (or atomic activities[4]).

The steps used to graphically depict the different sub-models are similar to the steps proposed in the previous phase for the high-level diagram. The following steps can be used to decompose processes on the high-level process model:

1. For each primary process, identify the set of affiliated sub-processes involved in the generation of the output resource(s).
2. For each sub-process, define the goal, input resources and output resources (similar to Phase 4).
3. Link the sub-processes with one another through input and output resources.
4. Draw the process model, which graphically depicts the sub-processes and their relationships between one another.
5. Repeat these steps for each of the sub-processes in the process model, until all sub-processes are atomic *or* the development team decides against further refinement.

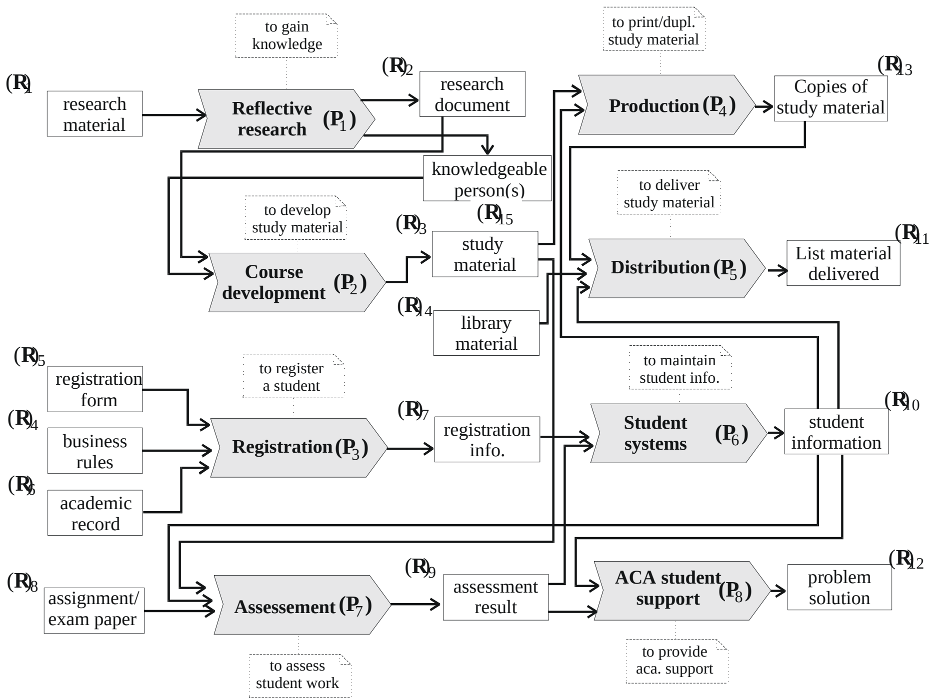
[4] An atomic activity is a process that cannot be broken down into further sub-processes.

Implementation at other institutions

The methodology was successfully put to the test at Unisa and process models were built for the core processes as well as selected sub-processes. We decided to test it at two other institutions with a different teaching model from the distance education model used at Unisa. A residential University and residential Technikon were selected as case studies. We used interviews as our primary information-gathering technique.

The results from the two case studies, using the proposed methodology, was a high-level process model, similar to the process model presenting the core processes at Unisa. The high-level process model developed at Unisa is presented in Figure 4.

Figure 4: High-level process model



(Cloete, Van der Merwe et al., 2003)

Both residential institutions argued that the *Production* and *Distribution* processes should be grouped together and should not be separate core processes. Lecturers at residential institutions have more responsibility for printing and distributing course material, in contrast to distance education institutions, which have formal units responsible for these tasks. The ideal scenario is that lecturing staff should not be responsible for administrative tasks such as copying course material, so from that angle we recognized the reasoning, but decided to keep the two processes as separate core processes.

We proceeded to extract the sub-processes (and further sub-processes) for the core processes. From this activity we learned that the processes within a core process are similar in the different environments, although the sequence of events may differ. For example, during the on-campus registration process at Unisa (for a new registration), a student will first receive a student number and then course enrolment verification is done. The residential Technikon first does the verification activity before the student number is issued.

Conclusion

One of our main reasons for embarking on the research effort was to suggest a method that will limit the time spent on the requirements elicitation procedure. In studying the first application domain, we spent nearly six months on acquiring a comprehensive understanding of it. We used our theoretical understanding of requirements elicitation and the experience gained from our environment to extract essential steps to simplify the requirements elicitation procedure in a specific application domain. The application of the methodology to other application domains resulted in a set of process models being constructed, in a significantly shorter period.

Acknowledgement

I wish to acknowledge the contributions of my supervisor, Professor Elsabe Cloete and co-supervisor, Professor Laurette Pretorius, who supported me throughout this project. I would also like to thank the administrators of the National Research Fund[5] in South Africa, whose co-operation made it possible for me to attend this conference.

References

ee, J and D Berry (2002). Requirements Elicitation Notes, Available from: http://www.student.math.uwaterloo.ca/~cs445/handouts/lectureSlides/F02/documents/elicitation.4.pdf.

Cloete, E, A Van der Merwe, et al. (2003). E-learning in a strategic educational institution. **To be published in 2003**.

Eriksson, Hans-Erick and Magnus Penker (2000). Business modeling with UML. New York, John Wiley & Sons.

Kononya, G and I Sommerville (1998). Requirements Engineering - Processes and Techniques. Chichester, John Wiley & Sons.

Maciaszek, L.A. (2001). Requirements Analysis and system design - Developing Information Systems with UML. Harlow, England, Addison-Wesley.

Porter, Michael (1985). Competitive Advantage - Creating and Sustaining Superior Performance. New York, The Free Press.

[5] This material is based upon work supported by the National Research Foundation under Grant number 2053850.

General comments

This paper is considerably over length. It needs to be cut back by nearly 50%. It also needs to be formatted correctly in places to conform with the authors guidelines.

The author has not addresssed the referees comments from the previous draft.