Proc. of the Second Intl. Conf. on Advances In Computing, Communication and Information Technology- CCIT 2014.

ISBN: 978-1-63248-051-4 doi: 10.15224/ 978-1-63248-051-4-38

Exploiting Enterprise Organograms to facilitate Goal/Requirements Elicitation.

Cyrille Dongmo, John Andrew Van der Poll

***Abstract* — Amongst other advantages, Goal-Oriented Requirements Engineering (GORE) methods facilitate the process of requirements elicitation and analysis, yet little is known about, e.g. the underlying paradigm relating goals to the structure and processes of the organisation. In this paper we postulate that since goals are inherently part of any organisation, and an information system aims to achieve specific organisational objectives, these ought to address a subset of goals that embody the main objectives of the organization or those that may be derived from them. The systemic paradigm has contributed much to enterprise architecting and modeling, both being vital inputs to goals and requirements elicitation. An enterprise architecture is, however, not readily available since its construction may be a costly exercise. In contrast, an enterprise organogram, which depicts a holistic view of the enterprise, may be easier to build. This paper proposes guidelines to exploit organograms to facilitate goals/requirements elicitation. Strategies to manipulate the proposed model in order to systematically identify potential sources of information are defined. A case study is used to illustrate our approach.**

***Keywords*— Enterprise model, Organogram, Goal elicitation, Requirements Engineering, Directed graph.**

I. Introduction

The need to produce sound models for enterprises has been expressed in the literature [19]. Research has been conducted to establish integrated models which may facilitate the understanding and analysis of enterprises, thereby improving on organizational performance and productivity [16, 19, 30].

Initially, research focused on areas such as: business processes, human resources, manufacturing, production, marketing, etc. Various (isolated) techniques, models and methods, e.g. business process models, manufacturing models and production models were defined.

Cyrille Dongmo
School of computing/ College of Science Engineering and Technology / University of South Africa (Unisa)
South Africa

John Andrew Van der Poll
Graduate School of Business Leadership (SBL) / University of South Africa (Unisa)
South Africa

This led to a multiplicity of individual information systems and software applications being developed for specific areas of the enterprise, bringing forward the need for integrating individual models to improve on the overall performance of the enterprise. Subsequently, Enterprise Modeling and Integration (EMI) (e.g. [16, 28, 30]) emerged. EMI is a comprehensive approach to enterprise modeling and analysis which involves: Enterprise Engineering (EE) [12], Enterprise Integration (EI) [19], and Enterprise Modeling (EM) [11, 34]. To our knowledge, no common or clear-cut definition for any of these concepts have yet been developed. However, collectively these techniques share a common objective which is to achieve market advancements in terms of overall organisational performance [28]. The use of the above concepts (EE, EI and EM) may benefit the following domains:

- ***Knowledge management***: This refers to an abstraction of the reality to facilitate the understanding of the enterprise, including its components (e.g. resources, processes and information). It is more likely represented as a model [31] hence, may be interpreted as a result of Enterprise Engineering or the Enterprise Modeling technique. The level of detail of the knowledge is relative to the level of abstraction of the reality.
- ***Resources:*** An example of enterprise resources is given in the 3M \& 1 system [16], where 3M \& 1 indicates: HuMan, Material/Machine, and Money, all examples of physical resources. These are some of the enterprise entities that have to be abstracted in a model [33].
- ***Business processes***: These are the main activities of the enterprise and have been researched since the early days of computing. In the literature, system or model integration refer mainly (but not only) to the integration of different business process models or systems, both internally and externally. The internal integration aims to establish the relationship between enterprise processes and resources, whereas the external integration attempts to establish a beneficial relationship with the influential elements from the environment.
- ***Environment:*** The environment of an enterprise refers to anything outside the enterprise that may directly or indirectly stimulate some changes to occur within the enterprise. It defines the conditions, e.g. geographical area and policies regulating the operations of the enterprise.

Proc. of the Second Intl. Conf. on Advances In Computing, Communication and Information Technology- CCIT 2014.

ISBN: 978-1-63248-051-4 doi: 10.15224/ 978-1-63248-051-4-38

- ***Internal relationships:*** These define the inter-dependencies between the entities within an enterprise.
- ***External relationships:*** These represent the inter-dependencies between the enterprise and its environment. Activities include: other enterprises (Business-to-Business - B2B) and organisations, governments (Business-to-Government - B2G), and market places (Business-to-Customer - B2C) [9].
- ***Decision making:*** This is a critical and continuous aspect of the enterprise, applied at upper organisational levels to facilitate the on-going strategic activities of the enterprise. A decision-making model is highly dependent on the knowledge and understanding of other aspects of the enterprise, including those above [12].

Each of these important aspects of enterprise architectures addresses directly or indirectly, at a specific level of abstraction, at least one of the six Zachman interrogatives or *Abstractions* [7], shown in Figure 1.

Figure 1 - Zachmann Primitive interrogations or Abstractions[7]

"What" Material Description	"How" Functional Description	"Where" Spatial Description	"Who" Operational Description	"When" Timing Description	"Why" Motivation Description
Structure (Things)	*Transform* (Processes)	*Flow* (Locations)	*Operations* (People)	*Dynamics* (Events)	*Motivation* (Strategies)

Thus, one of the ultimate goals of enterprise architectural modeling is to present a holistic view of an organisation to facilitate decision making that generally leads to changes. Since deciding on what should be computerised and why should it be computerised is an outmost decision to take within an enterprise, enterprise architecture remains a vital source of information and therefore, an important asset for goals and requirements elicitation and analysis [11]. The challenge is that Enterprise models and architectures are normally very large, complex structures that may not be constructed and manipulated easily. From Zachman's framework perspective [6, 35], a complete enterprise architecture comprises the description of each aspect of the organisation, responding to the interrogatives in Figure 1, at different levels of details:

(1) scope planning,
(2) business modelling as perceived by the owner,
(3) system modelling as perceived by the designer,
(4) technology modeling,
(5) details representations.

This paper proposes an extendable model, based on enterprise organogram, that presents in an hierarchical structure, each element of the enterprise associated with the objective(s) that the element aims to achieve. In any area of activity within the enterprise, the proposed model provides a means to represent each component and its sub-components, participating directly or indirectly to the life of the organisation. The model addresses, at the abstraction levels (1) and (2), three of the six primitive questions in Figure 1:

The ***What*** of the enterprise: it represents the structure of the enterprise in an organogram like diagram,

The ***Why*** of the enterprise: guidelines are provided to help complete the original organogram with business objectives,

The ***Where*** of the enterprise: guidelines are given to add resources and services to the original organogram.

We believe our proposed model to be useful in Goal/Requirement analysis activities since such an analysis is more concerned with the "What" and "Why" of the system being analysed.

II. Challenges in Goal - and Requirements elicitation

It is quite interesting to observe that ever since Brooks [4] brought forward the idea of the requirements elicitation and analysis being the hardest and most critical stage in software development, critics have been very rare. Instead, intensive research has been undertaken to improve on requirements elicitation techniques. However, despite tangible results obtained, e.g. a multiplicity of notations, the search for improved guidelines and appropriate processes remain challenging.

A. *Requirements elicitation*

Requirements elicitation in software development aims to uncover, create, and structure the needs of the users and other stakeholders, and communicate and discuss with other parties (e.g. development team, sponsors) to reach an agreement. The process is continuous and iterative in nature, since it is during the subsequent development phase that user needs are well understood and, therefore, susceptible to change. In practice, the main activities include [2, 17]:

(a) Preliminary data collection,
(b) Requirements gathering'
(c) Requirements evaluation and rationalization'
(d) Requirements prioritization'
(e) Requirements integration and validation.

Comprehensive requirements elicitation is the basis for constructing a good software specification that ought to be: correct, unambiguous, complete, consistent, ranked for important and/or stability, verifiable, modifiable, and traceable [10]. It is inherently a difficult activity since it involves people with differing goal sets and having different areas of expertise and cultural backgrounds. The main, commonly accepted problem areas include: *scope*, *understanding*, and *volatility* [17, 22, 32].

- ***Scope challenges*** involve the inability to establish the boundaries of the system to be built, delimiting both the domain and the environment of the system (potential

Proc. of the Second Intl. Conf. on Advances In Computing, Communication and Information Technology- CCIT 2014.

ISBN: 978-1-63248-051-4 doi: 10.15224/ 978-1-63248-051-4-38

source of requirements), as well as the diversity of the stakeholders. These problems may lead to incorrect requirements that suffer from incompleteness (with respect to the informal or semi-formal user requirements), redundancy, ambiguity, non-verifiability and non-utility. Establishing a solid understanding of the enterprise is emphasized by Michael and Kao [17]: "*The understanding of organizational, environmental, and project context thus provides a good starting point for requirements elicitation.*" The focus of this paper is on this challenge.

- Problems of ***understanding*** are attributed mainly to the communication latency between stakeholders with various backgrounds and different needs. The difficulties experienced by the analyst to comprehend the application domain, and challenges of the users and/or customers to understand the technical language of the analyst are illustrative. Such challenges often stem from the non-elicitation of tacit knowledge in the minds of stakeholders [8].
- ***Volatility*** results often from perceptions becoming clearer during the development process. Initial requirements are changed or new requirements are added to reflect the `evolving' needs of the customers. Volatility in requirements may also stem from political decisions, new managerial rules in an organisation and/or the inherent changing environment [13].

Various approaches have been proposed to improve on the process of requirements elicitation. However, as observed in [36], the inability of the proposed solutions to provide definitive guidelines is indicative of the complexities involved in requirements elicitation - sequences of activities often depend on specific project circumstances.

Bearing their own weaknesses, the goal-driven requirements engineering methods are currently among the promising and leading approaches. An overview of these methods is given next.

B. *Goal elicitation methods*

The concept of a goal refers to an objective that the envisioned system ought to achieve. Although traditional requirements engineering methods do not explicitly integrate the concept of goals, they nevertheless recognise the need to explicitly express the "why" of each requirement. An increasing interest to involve goal analysis in requirements engineering processes has led to a whole new approach of Goal-Oriented Requirements Engineering (GORE) [27]. The principal contributions of GORE are to achieve requirements completeness, avoid irrelevant requirements, clarify requirements to stakeholders, provide a natural mechanism to structure complex requirements, facilitate the choice of alternatives, manage conflicts among various viewpoints, manage requirements evolution by providing a means to separate stable from more volatile requirements, and more importantly, drive the identification of requirements to support them [29]. Requirements are generated from goals through goal refinement [36]. High level goals are progressively decomposed, using the "and" and "or" connectors to relate sub-goals to each other until a complete set of functional and non-functional requirements have been derived. As noted by [36].

"*In general, one of the risks when using goal based approaches is that errors in the high-level goals of the system made early on can have a major and detrimental follow on effect, and that changing goals are difficult to manage.*"

Such a risk shows the necessity to derive appropriate models, guidelines or protocols to facilitate the elicitation, the validation and verification of systems goals at an initial phase of requirements analysis. The need to derive such models is further supported, amongst others, by two realities. The first is that goals themselves ought to achieve higher-level enterprise objectives, which are inherently more business-based than IT related [21]. A Goal model provides a traceability link between high-level business strategies and the technical requirements [27]. The second observation relates to the complex nature of an enterprise [26].

C. *An enterprise as a complex system*

Different domain aspects have to be considered in enterprise modeling and architecting, and as noted in [33], a further difficulty results from a lack of solid theoretical foundations in the discipline of enterprise architecture. Hence, the building of models relies much on an architect's experience. That said, systems science is an emerging approach which may hold much promise for enterprise analysis and modeling [14, 25, 33].

Examples of the use of systems science are: the Merise method [18], which is one of the successful systemic software Engineering methods [24], the Total architecture [20], which is a purpose-oriented enterprise architecture that abstracts away from the mere IT-oriented modeling of an enterprise, and the Computer Integrated Manufacturing Opened System Architecture (CIMOSA) [12]. Similarly to other systems, the complexity of an enterprise is attributed mainly to the various aspects, introduced in Section I, that are involved in the study of the enterprise, the relationships among those elements, and the inherent dynamic nature of an enterprise in a changing environment.

Keeping in mind that *scope definition* is one of the harder challenges in requirements engineering, Goal-Based requirements analysis, generally, assumes that goals are not, a priori, documented explicitly. It is, therefore, the responsibility of the requirements engineer to explore various sources of information available to identify, create and organise goals [1]. The sources of information to be explored include: stakeholders, policies, transcripts, work flow diagrams, requirements, mission statements, corporate goals and

Proc. of the Second Intl. Conf. on Advances In Computing, Communication and Information Technology- CCIT 2014.

ISBN: 978-1-63248-051-4 doi: 10.15224/ 978-1-63248-051-4-38

interview facts. Techniques to perform such exploration have been proposed by [23]:

- Understanding stakeholders' problems and negating them.
- Extracting intentional statements from stakeholders: interview transcripts, enterprise policies, enterprise mission statements, enterprise goals, workflow diagrams, and scenarios.
- Asking "How" and "Why" questions about these initially identified goals in order to move about the goal hierarchy.
- Asking "How else" questions to identify alternative goals.

The above guidelines are complemented with heuristics to facilitate the identification of goals from a given source, and address separately goals per type. However, with the increasing complexities of enterprises and software systems, the scope definition problem remains.

Enterprise modeling/architecting is known to be appropriate tools for requirements engineering [11], yet with various models that may be built to represent the different facets of an enterprise (e.g. functional, informational, resource and organisational [5]), the establishment of appropriate guidelines/pointers to reference relevant sources of information (or the specific areas within the enterprise where needed information can be found) remains challenging.

In our work we propose the use of an organogram to address the above scope delimitation challenges and as illustration, we employ the following case study:

III. Case study

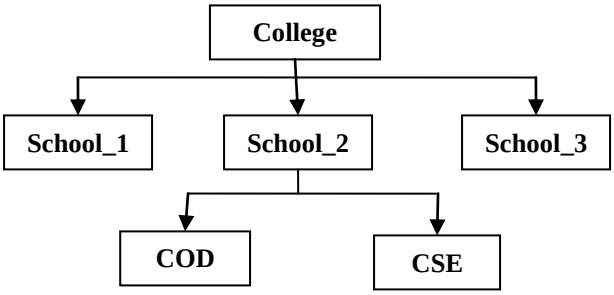
A. *University College*

A university college is considered in this work for illustration (see Figure 2). Although the organogram in Figure 2 does not represent any real-world institution, it purposes to depict some common structural aspects of real colleges. The college is composed of 3 schools, namely, School_1, School_2, and School_3. School_2 is decomposed into COD and CSE where COD refers to the Chair of Department, and CSE stands for Center for Software Engineering.

B. *Software for quality assurance in School_2*

Let's consider an IT project initiated at School_2 to provide the <<Quality assurance>> (Figure 3) sector with a software tool. We postulate that if **Obj_S2Q** is the set of business objectives for the "Quality Assurance" within School_2, the set of high-level objectives of the IT tool to be produced is inherently a sub-set of Obj_S2Q. Given the objectives of the software to be developed, we need to systematically identify all the nodes within the college's organogram (in Figure 3) that ought to be considered during the phase of goal/requirement analysis, as well as the business objectives associated with each node that require analyst attention.

Figure 2 - College organogram



I. The proposed approach

Any enterprise - be it private, public, a government institution or an NGO aims to achieve some high level objectives.
Such objectives are realised by using enterprise resources to render services or perform regular or occasional activities. As noted in [25]:

"*An enterprise is a goal-directed organization of resources: human, information, financial, and physical, and activities, usually of significant operational scope, complication, risk, and duration*".

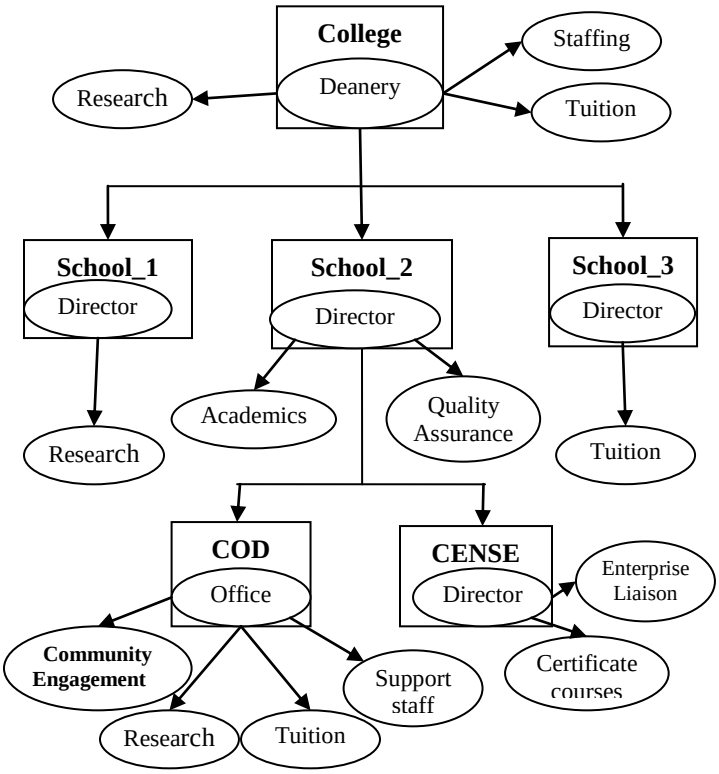
An enterprise is generally organised into domains or sub-structures (e.g. Human Resources, Research, faculties, schools, departments, etc.) where some enterprise resources are consumed to achieve sub-objectives. Generally, large enterprises are structured into multiple hierarchical levels, with the operational domains towards the bottom of the structure, and the decisional (managerial) entities at the top. For example, in an academic institution, one may have: Top-management, Institutional management, faculties, schools, departments, etc. Figure 3 is an updated version of the organogram in Figure 2 to which operational components were added.

At college level, School_2 is viewed as an operational entity, supporting the relevant college objectives. At school level, the operational entities may be diverse and overlapping, e.g. a research committee, academic staff members, a tuition committee, etc. Each of these ought to support the objectives of the school, which are in fact sub-objectives of the college. For example, any school research objective is a sub-objective of a research objective of the college. Achieving such college research objective is facilitated through realising the relevant research objectives of the schools and departments within the college.

Proc. of the Second Intl. Conf. on Advances In Computing, Communication and Information Technology- CCIT 2014.

ISBN: 978-1-63248-051-4 doi: 10.15224/ 978-1-63248-051-4-38

Figure 3 - College organogram with operational components



This work also suggests that along the hierarchical organisation of an enterprise, decision points are the places where the objectives and sub-objectives of the enterprise may be decided on, documented, assigned to the appropriate agents, and monitored. In the updated organogram of Figure 3, the dean's office (deanery) and the Director's office in each school are the decision components of the college - the Director's office is responsible for sub-objectives delegated from the dean's office.

A decisional body also constitutes a connection point with the outside world that is, the environment. For example, the office of the director of School_2 may sign research/training MOAs (memoranda of agreements) with industry.

In general, ICT projects aim to produce operational resources to reinforce the existing processes, services or activities of an enterprise in order to contribute to the achievement of some business objectives or sub-objectives of the enterprise. Such projects are therefore rooted and initiated at the operational levels of the enterprise with the purpose to contribute to the resolution of managerial problems. For example, faster banking software may be required to improve on customer service and throughput. In a similar vein, the software may be needed at the cashier for data mining of customer information for marketing purposes.

To address the challenge of goal elicitation and analysis, we suggest two complementary approaches: bottom-up and top-down.

(1) The ***bottom-up*** approach identifies all decisional components to be included, as well as the operational entities involved at each decisional point, thereby delimiting the scope of the analysis.

(2) The ***top-down*** approach addresses the implementation of the prescribed guidelines for goal analysis: interviews, documents and analysis, etc. This is often performed by refining abstract objectives from the highest decisional points down to the operational elements of the enterprise.

Traditionally, Goal-Oriented Requirements Engineering methods, employing consistent guidelines and heuristics follow a top-down approach [23], hence the focus of our work is on the bottom-up approach.

C. *The bottom-up approach*

As mentioned earlier, this approach is based on the following viewpoints:

a) The organisation

An enterprise can be organised in multiple hierarchical levels, where each abstract level represents a sub-structure or a domain composed of operational elements and a decisional body. The operational elements are the active components of the enterprise, which aim to achieve business objectives under the responsibility of the decisional body they are associated with. The decisional body, commonly known as the management, is responsible for managerial activities and the coordination and evaluation of the activities of the operational components reporting to it. The decisional body is responsible for the achievement of a given set of business objectives, and liaises with higher level structures, as well as the environment.

b) The envision system

It may be argued that a software project is always rooted to at least one operational component of the enterprise, since such a project aims to provide tool(s) to facilitate the operations of the enterprise. As described above and illustrated in Figure 3, at each level of abstraction within an hierarchical organisation, there is a decisional body and at least one operational component. Since a decisional body is itself an operational component with managerial/decisional responsibilities, a software project can be initiated to support the activities of the enterprise at any given level.

c) The objectives of the system

The high level objectives of a software system are therefore derived from the business objectives of the operational elements of the enterprise. Naturally, the system's objectives at this level of abstraction are those that will be refined, during the top-down phase, into (operational) goals justifying the functional and non-functional requirements of the system.

Consequently, the high-level objectives of a system are subject to purpose of the operational elements being computerised, the

Proc. of the Second Intl. Conf. on Advances In Computing, Communication and Information Technology- CCIT 2014.

ISBN: 978-1-63248-051-4 doi: 10.15224/ 978-1-63248-051-4-38

relationship between these enterprise's elements and other elements within the same domain (horizontal relationship), and the relationship between these elements and other enterprise elements at higher and lower levels in the organisation (vertical relationship). These two relationships are based, for example, on the interdependencies of the objectives each of the operational elements ought to achieve.

- The horizontal relationship analysis, will aim to identify within the same domain, those operational and/or decisional components that need to be considered.
- The purpose of the vertical relationship analysis is to identify, at both the higher and lower levels of the hierarchical organisation of the enterprise, operational and/or decisional components that need to be included in the investigation.

d) The source of information

In practice, business objectives, the purpose of services and resources of the enterprise are generally not listed or directly accessible from the environment. Instead, they have to be extracted from the mission statement, policies and other documents regulating the functioning of the enterprise.

Naturally, these documents are sometimes difficult to localise. When analysing the objectives of an enterprise, important information are mostly accessible through people, and the difficulty is to identify such individuals.

From the above premises we have:

(1) The immediate source of information includes the operational elements to which the system is rooted.

(2) Operational components under the same decisional body may be included on the ground level resulting from a horizontal relationship analysis.

(3) Operational elements at higher or lower levels in the organizational hierarchy may be included depending on the outcome of the vertical relationship analysis.

The information that may be used and the proposed modeling of such information are presented next.

D. The modeling

a) Assumptions

Our model to guide scope definition is founded on two observations: First, an enterprise is naturally objective-oriented, and uses resources to perform activities or render services in order to achieve its business objectives. Hence, the three types of information retained for modelling purposes are: ***objectives***, the ***services*** and ***resources***. The term service is used to describe any activity, operation or task to be performed.

Figure 4 - Decomposing a business objective

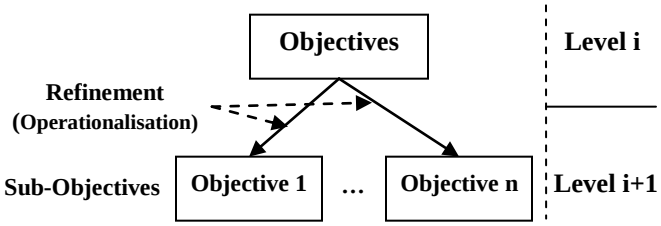


Figure 4 shows how high-level business objectives are refined to sub-objectives, progressively down to the operational levels.

The second assumption is that most companies have a hierarchical structure with shared responsibilities throughout the breadth and depth to meet the objectives.

Even though an enterprise organogram may not be explicitly documented, it should be reasonably easy to reproduce since it ought to picture a holistic representation of the enterprise at a specific point in time. However, the difficult part is to clearly define the responsibilities of each sub-structure, since the college's objectives (as is the case with any enterprise) are progressively divided into smaller and more realisable sub-objectives, assigned to smaller domains (sub-structures) within the college. Defining the objectives of a company's sub-domains is normally not the responsibility of a requirements engineer, since these are defined when creating the sub-domains. If the objectives are not documented, the engineer would have to identify them.

Some of the main advantages of an organogram stem from the fact that the diagram represents, from a specific view point, the entire company, thereby favouring a systemic analysis. The information is lightweight since details are not considered yet and updating (e.g. slight restructuring) and exploration of the diagram in search of information is facilitated.

b) Construction of the model

The construction process assumes the availability of an existing organogram, otherwise one is created. The organogram is populated and completed with decisional and operational elements (see Figure 3). The steps to follow are:

Guidelines for constructing the organogram

(1) Create an organogram for the company, based on business objectives, if no organogram is found.

(2) Transform the organogram to add, at each level, one decisional component to each domain (sub-structure) at that level.

(3) Transform the organogram to include, at each level, operational elements to each domain or sub-domain. Such operational elements should be characterised mainly by the service(s) or operations they perform.

Proc. of the Second Intl. Conf. on Advances In Computing, Communication and Information Technology- CCIT 2014.

ISBN: 978-1-63248-051-4 doi: 10.15224/ 978-1-63248-051-4-38

(4) At each level in the organogram and for each decisional component, determine the objectives assigned to such component.

(5) For each operational element, when possible, relate the objectives to other elements within the same domain (horizontal relationship) and to the operational component in the immediate higher-level domain (vertical relationship).

E. *Concepts/Theory*

Since an organogram has a directed graph structure, graph concepts are considered for the modeling. This has the advantage that existing graph theory on modeling and graph search may be exploited. The definitions, presented next and further graph theory that may be utilized in later stages are taken from [3].

Definition 1. A directed graph is a pair G = (V, E) where V is a set whose elements are called vertices or nodes, and E is a set of ordered pairs of elements of V. Elements of V are called arcs or directed edges.

Let $x \mapsto y \in E$, x is called the tail of the edge or the direct predecessor of y, whereas, y is called the head of the arc or the direct successor of x.

Definition 2. A sub-graph of a graph G = (V, E) is a graph G' = (V', E') such that $V' \subseteq V$ and $E' \subseteq E$.

Traversing graphs. Problems on a graph generally require that each vertex and each edge of the graph be reached and examined. As noted in [3], breadth-first search and depth-first search are two fundamentals strategies that provide an efficient way to "visit" each vertex and edge exactly once.

Based on these two graph traversal strategies, this work seeks to derive or adapt a mechanism whereby, from a list of sub-objectives of an operational or a decisional node, the adapted strategy should be able to search the entire organogram and produce the list of nodes and associated objectives that need to be considered during Goal/Requirements elicitation phase.

Next the graph model of the organogram in Figure 3 is presented to illustrate the modeling approach.

F. *Graph model of the organogram in Figure 3*

A node of the graph is either a decisional element or an operational one. Table 1 presents the labeling / codification of the different nodes of the organogram. Research objectives at the college level are achieved through the research activities at the deanery, as well as the achievement of research objectives at School_1 and 2.

Table 1 - Labeling the nodes of Figure 3

Node label (identifier)	Description of each node		
	Domain	***Management***	***Operational element***
CD	College	Deanery	Dean's office
CR	Colllege	Deanery	Research
CS	College	Deanery	Staffing
CT	College	Deanery	Tuition
S1	College	School_1	Director office
S2	College	school_2	Director's office
S3	College	school_3	Director's office
S1R	School_1	Director's office	Research
S2A	School_2	Director's office	Academics
S2Q	School_2	Director	Quality assurance
S2D	School_2	COD	Office
S2C	School_2	CSE	CENSE office
DCE	School_2	COD	Community Engagement
DR	School_2	COD	Research
DT	School_2	COD	Tuition
DS	School_2	COD	Support staff
CEC	School_2	CSE	Certificate courses
CEE	School_2	CSE	Enterprise liaison
S3T	School_3	Director	Tuition

A graph model of the organogram is therefore given by the sets nodes V_{fig3} and the set of edges E_{fig3}.

$$V_{fig3} = \left\{ \begin{matrix} CD, CR, CS, CT, S1, S2, S3, S1R, S2A, S2Q, \\ S2D, S2C, DCE, DR, DT, DS, CEC, CEE, S3T \end{matrix} \right\}$$

and

$$E_{fig3} = \left\{ \begin{matrix} CD \mapsto CR, CD \mapsto CS, CD \mapsto CT, CD \mapsto S1 \\ CD \mapsto S2, CD \mapsto S3, S1 \mapsto S1R, S2 \mapsto S2A \\ S2 \mapsto S2Q, S2 \mapsto S2D, S2 \mapsto S2C, S2D \mapsto DCE, \\ S2D \mapsto DR, S2D \mapsto DT, S2D \mapsto DS, S2C \mapsto CEC, \\ S2C \mapsto CEE, S3 \mapsto S3T \end{matrix} \right\}$$

This simplified model of the organogram, although does not incorporate information on nodes such as: the list of objectives, the physical location, resource allocated, etc., can be flexibly manipulated: it may benefit from sets operations and existing graph search algorithms.

Proc. of the Second Intl. Conf. on Advances In Computing, Communication and Information Technology- CCIT 2014.

ISBN: 978-1-63248-051-4 doi: 10.15224/ 978-1-63248-051-4-38

G. Node identifier and Objectives

An identifier is used to uniquely reference each node. For example, the first column of Table 1 gives the set of all identifiers of nodes of the organogram in Figure 3. In a Z like notation [15], an identifier, as well as a business objective, is considered a given set:

$$[Identifier, Objective] \quad (1)$$

The graph model of the organogram uses only the identifiers of nodes since other information may be associated to the node and accessed by linking them to the identifiers by means of relations or functions.

H. Linking objectives to nodes

As mentioned above, objectives and other information (e.g. services and resources) may be associated to nodes by means of a relation.

$$nodeObj: Identifier \leftrightarrow Objective \quad (2)$$

The relation *nodeObj* associates to each identifier the business objectives of the node that it identifies.

(1) The list of objectives at a node can be generated by applying successively the domain restriction [1] and the range operators to the function *nodeObj* as shown in the following formula:

$$\text{ran}(\{nodeid\} \lhd nodeObj) \quad (3)$$

In (3), *nodeid* is the identifier of the node.

(2) Adding an objective (obj) to a node is performed by means of set-theoretic union. Similarly, removing an objective from a node maybe performed by means of set subtraction.

I. Relationships between objectives

We consider two important relationships, namely, the horizontal and vertical relationships. The horizontal relationship models the dependencies between the objectives at the same abstraction level. E.g. Quality assurance may be performed in School_2 on all ready to publish articles to ensure that Academics produce high quality publications.

$$hRel: Objective \leftrightarrow Objective \quad (4)$$

The vertical relationship maps a sub-objective to the higher level objective from which it was obtained by refinement/decomposition.

$$vRel: Objective \leftrightarrow Objective \quad (5)$$

[1] [1] Domain restriction operator restricts a relation to that part where the domain is contained in a particular set

J. Search algorithm

Considering that a software project can be initiated for any operational or decisional component, our problem presents two specifics:

- The search may start from any node of the graph with an initial list of objectives.
- Further search highly depends on the current list of discovered objectives and nodes processed on the basis of horizontal and vertical relationships between the objectives.

The second aspect suggests a dynamic programming approach. E.g. if none of the discovered objectives is involved in any of the two relationships with a non-discovered objective, the search stops. Owning to the fact that the depth-first and breath-first traversal algorithms are naturally recursive which can therefore be transformed into dynamic algorithm in case of inefficiencies, we choose to base our strategies on them.

Algorithm 1.0

Input *E, CurV, CurObj, CurHrel, CurVrel*

Output *CurV', CurObj', CurHrel', CurVrel',*

Initialize (CurV, CurObj, CurHrel, CurVrel)

While *(there are vertex v in CurV not coloured black)*

For Each *vertex v in CurV*

If *color(v) is white then*

'horizontal processing of v

Apply Algorithm 1.1

Change v color to grey

ElseIf *Color(v) is grey then*

'vertical processing of v

Apply Algorithm 1.2

Change color of v to black

End if

Next *vertex*

Wend

End

E is the set of directed edges; *CurV* contains the currently identified vertices that may be considered during requirements elicitation. *CurObj* is the set of objectives so far identified and *CurHrel* and *CurVrel* represents, respectively, the horizontal and vertical relationships between the currently identified objectives. The initialisation of CurV colors each vertex white. When a vertex is first visited, the horizontal search is performed and the vertex is colored grey.

The horizontal analysis: For a given node, the purpose is to identify those objectives of the node that are in a horizontal relationship with the currently identified objectives.

Proc. of the Second Intl. Conf. on Advances In Computing, Communication and Information Technology- CCIT 2014.

ISBN: 978-1-63248-051-4 doi: 10.15224/ 978-1-63248-051-4-38

Algorithm 1.1 – Horizontal search

Input *V, CurObj, CurHrel*

Output *CurObj, CurHrel updated*

ListObjectives **=** ***ran*** *curObj*

For Each *Ov in ran ({v}◁ nodeObj)*

For each O in ListObjectives

If *Ov↦O ∈ hRel or O↦Ov ∈ hRel then*

Add Ov to CurObj

Add Ov↦O or O↦Ov to CurHrel

End if

Next O

Next Objective

The vertical analysis: For the input node, the main purpose is to identify direct predecessors and successors of the node which objectives are in vertical relationship with the objectives of the input node.

Algorithm 1.2 – Vertical search

Input *v, CurObj, CurV, E, CurVrel*

Output *CurV, CurObj,CurVrel updated*

ListObjectives **=** *ran ({v}◁ nodeObj)*

**** analysing the direct predecessor of v ****

If *v has a direct predecessor w (w↦v∈E)* ***then***

If *at least one objective of w is in vertical relationship with at one or more objectives of v* ***then***

Add the vertex w to curV if not yet added

Add each Ow ↦ *Ov* of vRel to CurVrel, where Ov is an objective of v and Ow objective of w.

Add each Ow to CurObj

End If

End If

**** analysing the direct successors of v ****

If *v has at least one direct successor* ***then***

For each *direct successor s of v (v↦s ∈* E)

If *an objective Os of s is in vertical relationship with Ov (Ov↦Os ∈ Vrel)* ***then***

Add s to CurV is not yet added

Add Os to CurObj

Add each Op ↦ *Ov* of vRel to CurVrel where Ov is an objective of v and Os objective of s.

End If

Next *successor*

End If

IV. Illustration

The follow-up of this work, now under consideration, proposes an Object-Z specification of the above organogram model and the formalisation of the algorithms. The formal specification was animated using Prolog whereby we successfully implemented the two important search operations and executed them to generate the expected outputs.

V. Conclusion and Future work

This paper proposed an approach for establishing guidelines to construct enterprise organograms in a bottom-up fashion and transform these into useful models that can be exploited in goal and requirements elicitation phases to identify vital sources of information within an entire organisation. Our approach also proposes strategies to manipulate the model and derive the necessary information in a simplistic manner. The main advantage of the proposed method stems from the simplicity and availability of enterprise organograms to which appropriate information may be cautiously added to construct flexible and lightweight enterprise models vital to goal and requirement elicitation.

An extension of this work, under consideration, proposes a formal specification with Object-Z of the organogram models and the manipulation strategies proposed in this paper. The Object-Z specification was animated using Prolog to establish the validity of the method. Since enterprises are becoming increasingly complex structures, an immediate consideration shall be to derive distributed versions of the proposed models and algorithms and means to adapt them to cloud computing.

References

[1] Annie I. Anton and Colin Potts. The use of goals to surface requirements for evolving systems. In *International Conference on Software Engineering*, pages 157–166, Kyoto, April 1998. URL http://-ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=671112.

[2] Gabriela Aranda, Aurora Vizcaino, Alejandra Cechich, and Mario Piattini. Strategies to minimize problems in global requirements elicitation. *CLEI Electron. J.*, 11 (1), 2008. URL http://dblp.uni-trier.de/-db/journals/cleiej/cleiej11.html#ArandaVCP08.

[3] Sara Baase. *Computer algorithms: introduction to design and analysis*. Addison-Wesley, 2000.

[4] Frederick P. Brooks. No Silver Bullet: Essence and Accidents of Software Engineering. *Computer*, 20 (4): 10–19, 1987. ISSN 0018-9162. doi: http://dx.doi.org/10.1109/MC.1987.1663532.

[5] Dursun Delen, Nikunj P. Dalal, and Perakath C. Benjamin. Integrated modeling: the key to holistic understanding of the enterprise. *Commun. ACM*, 48: 107–112, April 2005. ISSN 0001-0782. doi: http://doi.acm.org/10.1145/1053291.1053296. URL http://doi.acm.org/-10.1145/1053291.1053296.

[6] Zachman framework url. http://www.zachman.com/about-the-zachman-framework, 2014. Accessed July 18, 2014.

[7] David S Frankel, Paul Harmon, Jishnu Mukerji, James Odell, Martin Owen, Pete Rivitt, Mike Rosen, and Richard Mark Soley. The zachman framework and the omg's model driven architecture. 2003. URL **Error! Hyperlink reference not valid.**.

[8] W. Friedrich and John A. van der Poll. Towards a methodology to elicit tacit domain knowledge from users. *Interdisciplinary Journal of Information, Knowledge and Management (IJIKM)*, 2: 179 – 193, 2007. ISSN Print 1555-1229. URL www.ijikm.org.

ISBN: 978-1-63248-051-4 doi: 10.15224/ 978-1-63248-051-4-38

[9] S. Haag and M. Cummings. *Management Information Systems for the Information Age*. McGraw-Hill International Edition, New York, NY, 9 edition, 2013.

[10] IEEE Std 830-1998. IEEE Recommended Practice for Software Requirements Specifications. Technical report, IEEE, 1998. URL http://-ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=720574.

[11] Marite Kirikova and Janis A. Bubenko. Enterprise modelling: Improving the quality of requirements specifications, 1994.

[12] K Kosanke, F Vernadat, and M Zelm. Cimosa: enterprise engineering and integration. *Computers in Industry*, 40 (2–3): 83 – 97, 1999. ISSN 0166-3615. doi: 10.1016/S0166-3615(99)00016-0. URL http://-www.sciencedirect.com/science/article/pii/S0166361599000160.

[13] G.P. Kulk and C. Verhoef. Quantifying requirements volatility effects. *Science of Computer Programming*, 72 (3): 136 – 175, 2008. ISSN 0167-6423. doi: http://dx.doi.org/10.1016/j.scico.2008.04.003. URL http://www.sciencedirect.com/science/article/pii/S0167642308000464.

[14] M. N. Lakhoua. Investigation in the cooperation of systemic methods: Case study of an industrial process. *Scientific Research and Essays*, 6 (7): 1507–0513, 2011. ISSN 1992-2248. URL http://-www.academicjournals.org/SRE.

[15] David Lightfoot. *Formal Specification Using Z*. Grassroots Series. Palgrave, 2nd edition, 2001.

[16] Masayuki Matsui. Enterprise modeling and integration: A stochastic management approach. *International Journal of Production Economics*, 122 (1): 485–491, 2009. URL http://EconPapers.repec.org/-RePEc:eee:proeco:v:122:y:2009:i:1:p:485-491.

[17] Christel Michael and Kang Kyo. Issues in Requirements Elicitation (CMU/SEI-92-TR-012). Technical report, Software Engineering Institute, Carnegie Mellon University, 1992. URL http://-www.sei.cmu.edu/library/abstracts/reports/92tr012.cfm.

[18] D. Nanci. *Ingénierie des systèmes d'information avec Merise: vers une deuxième génération*. Collection Performance. Sybex, 1993. ISBN 9782736112912. URL http://books.google.co.za/-books?id=pbDqPQAACAAJ.

[19] A. K. Patankar and Adiga Sadashiv. Enterprise integration modelling: a review of theory and practice. *Computer Integrated Manufacturing Systems*, 8: 21–34, July 1995. ISSN 0951-5240/95.

[20] Brown Paul, C. *Implementing SOA: Total Architecture in Practice*. Addison-Wesley, 1st edition, 2008.

[21] Dick Quartel, Wilco Engelsman, Henk Jonkers, and Marten van Sinderen. A goal-oriented requirements modelling language for enterprise architecture. In *Proceedings of the IEEE International Enterprise Distributed Object Computing Conference, EDOC '09*, pages 3–13, Los Alamitos, CA, USA, 2009. IEEE Computer Society Press. URL http://doc.utwente.nl/67577/.

[22] Prasad Rajagopal, Roger Lee, Thomas Ahlswede, Chia-Chu Chiang, and Dale Karolak. A new approach to software requirements elicitation. In *Proceedings of the Sixth International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing and First ACIS International Workshop on Self-Assembling Wireless Networks*, SNPD-SAWN '05, pages 32–42, Washington, DC, USA, 2005. IEEE Computer Society. ISBN 0-7695-2294-7. doi: 10.1109/SNPD-SAWN.2005.5. URL http://dx.doi.org/-10.1109/SNPD-SAWN.2005.5.

[23] Gil Regev and Alain Wegmann. Where do goals come from: the underlying principles of goal-oriented requirements engineering. In *Proceedings of the 13th IEEE International Conference on Requirements Engineering*, pages 253–362, Washington, DC, USA, 2005. IEEE Computer Society. ISBN 0-7695-2425-7. doi: 10.1109/RE.2005.80. URL http://dl.acm.org/-citation.cfm?id=1099549.1100644.

[24] Arnold Rochfeld. Merise: An Information System Design and Development Methodology, Tutorial. In Stefano Spaccapietra, editor, *ER*, pages 489–528. North-Holland, 1986. ISBN 0-444-70255-5.

[25] William B. Rouse. Enterprises as systems: Essential challenges and approaches to transformation. *Syst. Eng.*, 8 (2): 138–150, June 2005. ISSN 1098-1241. doi: 10.1002/sys.v8:2. URL http://dx.doi.org/10.1002/-sys.v8:2.

[26] Oscar A. Saenz and Florida International University. *Framework for Enterprise Systems Engineering*. Florida International University, 2005. ISBN 9781109954913. URL http://books.google.co.za/-books?id=DL1itwAACAAJ.

[27] A. M. Sen and K. Hemachandran. Goal requirement engineering: A literature survey. *Assam University Journal of Science & Technology: Physical Sciences and Technology*, 6 (II): 16–25, 2010. ISSN 0975-2773 (Print). URL http://www.inflibnet.ac.in/ojs/index.php/AUJSAT/article/-view/176.

[28] Huat Lim Soon, Neal Juster, and Alan de Pennington. Enterprise modelling and integration: a taxonomy of seven key aspects. *Computers in Industry*, 34: 339–359, 1997.

[29] Axel Van Lamsweerde. Goal-Oriented Requirements Engineering: A Guided Tour. In *RE '01: Proceedings of the Fifth IEEE International Symposium on Requirements Engineering*, page 249, Washington, DC, USA, 2001. IEEE Computer Society.

[30] F. Vernadat. *Enterprise Modelling and Integration: Principles and Application*. Chapman & Hall, London, 1996.

[31] F.B. Vernadat. Enterprise modeling and integration (emi): Current status and research perspectives. *Annual Reviews in Control*, 26: 15–25, 2002. URL http://www.elsevier.com/locate/arcontrol.

[32] Jaya Vijayan and G. Raju. Requirements elicitation using paper prototype. In Tai-hoon Kim, Haeng-Kon Kim, MuhammadKhurram Khan, Akingbehin Kiumi, Wai-chi Fang, and Dominik ÅšlÄ™zak, editors, *Advances in Software Engineering*, volume 117 of *Communications in Computer and Information Science*, pages 30–37. Springer Berlin Heidelberg, 2010. ISBN 978-3-642-17577-0. doi: 10.1007/978-3-642-17578-7_4. URL http://dx.doi.org/10.1007/978-3-642-17578-7_4.

[33] Prof Alain Wegmann. On the systemic enterprise architecture methodology (seam). In *SEAM). Published at the International Conference on Enterprise Information Systems 2003 (ICEIS 2003*, pages 483–490, 2003.

[34] Nel Wognum. Enterprise modelling and system support. *Advanced Engineering Informatics*, 18 (4): 191–192, 2004. URL http://-doc.utwente.nl/76266/.

[35] John Zachman. The zachman framework for enterprise architecture. *Zachman International*, 2002.

[36] Didar Zowghi and Chad Coulin. *Requirements Elicitation: A Survey of Techniques, Approaches, and Tools*, pages 19–46. Springer, 2005. doi: 10.1007/3-540-28244-0_2. URL http://dx.doi.org/10.1007/3-540-28244-0_2.