# NLP-KAOS for Systems Goal Elicitation: Smart Metering System Case Study

Erik Casagrande, *Member, IEEE*, Selamawit Woldeamlak, Wei Lee Woon, *Member, IEEE*, H. H. Zeineldin, *Senior Member, IEEE*, and Davor Svetinovic, *Member, IEEE*

**Abstract—**This paper presents a computational method that employs Natural Language Processing (NLP) and text mining techniques to support requirements engineers in extracting and modeling goals from textual documents. We developed a NLP-based goal elicitation approach within the context of KAOS goal-oriented requirements engineering method. The hierarchical relationships among goals are inferred by automatically building taxonomies from extracted goals. We use smart metering system as a case study to investigate the proposed approach. Smart metering system is an important subsystem of the next generation of power systems (smart grids). Goals are extracted by semantically parsing the grammar of goal-related phrases in abstracts of research publications. The results of this case study show that the developed approach is an effective way to model goals for complex systems, and in particular, for the research-intensive complex systems.

**Index Terms—**Requirements engineering, goal elicitation, NLP, data mining, bibliometrics

## 1 Introduction

This paper presents an approach to facilitate goal elicitation from research publications. The approach integrates data mining and natural language processing (NLP) techniques with KAOS as a representative goal-oriented requirements engineering (GORE) method [1], [2], [3], [4]. These techniques are used to extract relevant goal-related information for the complex system under investigation from a large volume of literature in an automatic fashion. A need to analyze the vast number of legacy requirements specifications, research publications, standards, etc., is common for the analysis and development of complex systems. Data mining techniques can help reduce the time and effort spent in analyzing these resources, and data mining is applicable in a number of requirements engineering (RE) scenarios: analysis of legacy documentation, outdated requirement specifications, user documentation, documentation of competitor systems, etc.

Despite the applicability of data mining to various RE scenarios, in this paper, we apply our approach to the goal specification of a complex system in its early stage of the research and development. We refer to such systems as research intensive complex systems (RICS) in this paper.

Software engineering is often concerned with RICS. Currently, RICS include such systems as crowdsourcing systems, smart meters and smart grids, various kinds of intelligent self-aware systems, etc. Research itself is the origin of different innovations and advancements for systems, and a cost effective way of analyzing research output can help gain market competitiveness and stakeholder satisfaction. The critical step is getting the goals right. Therefore, identifying goals early for RICS may increase the potential of success and sustainability of the developed system. However, the challenge is that the RICS often do not have a particular document stating the problem and the issues which need to be addressed, and the main sources of information are academic, governmental and organizational publications and white papers. Often, there is a large number of publications available and their manual analysis is time and resources consuming.

Thus, this paper presents the computational and exploratory stages of extracting goals from textual data using our data mining and NLP approach integrated with KAOS. We refer to our approach as NLP-KAOS in this paper. In particular, we show how NLP-KAOS can produce a goal specification in a semi-automatic and iterative fashion. The computational core steps of NLP-KAOS approach can be summarized as follows:

- *Bibliometric data collection.* Abstracts of research publications are automatically collected from online databases for data mining. Abstracts are short but accurate descriptions of the state-of-art of innovative research in RICS.
- *Goal extraction from text.* Goals are extracted from the abstracts using NLP method. The method searches for sentences in the text which contain goal-related keywords. The Stanford Parser [5], [6] is used to tag semantic structures in these sentences. Goals from the parsed semantic trees are extracted using our rule-based algorithm.
- *Goals taxonomy creation.* A method for creating, managing and visualizing taxonomies of extracted goals is developed. A taxonomy is a particular form of domain knowledge representation that describes the hierarchical relationships between semantic concepts

- *The authors are with the Department of Electrical Engineering and Computer Science, Masdar Institute of Science and Technology, Abu Dhabi, UAE. E-mail: dsvetinovic@masdar.ac.ae.*

*Manuscript received 7 June 2013; revised 16 June 2014; accepted 10 July 2014. Date of publication 15 July 2014; date of current version 17 Oct. 2014.*
*Recommended for acceptance by S. Clarke.*
*For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.*
*Digital Object Identifier no. 10.1109/TSE.2014.2339811*

[7]. The construction of taxonomies from extracted goals is performed using a modified version of the Heymann Algorithm [8].

Following the goal taxonomy creation, KAOS is applied in the normal way starting with the automatically generated list of goals. This is the manual part of the NLP-KAOS approach.

In addition to the NLP-KAOS, the secondary contribution of this paper is the case study demonstration that goal analysis of research publications provides useful goal models for RICS. The smart metering system, a particular example of RICS system, is used to evaluate the NLP-KAOS approach.

Section 2 presents relevant background and related work on smart metering system, GORE, and NLP. Section 4 presents NLP-KAOS method and data acquisition. Section 5 presents the results of NLP-KAOS applied to the smart metering system case study. Section 6 presents the evaluation of the case study results. Section 7 presents the case study limitations and validity threats. Section 8 presents the conclusions and future work.

## 2 Background and Related Work

### 2.1 Goal-Oriented Requirements Engineering

A brief summary of the concepts of KAOS relevant to our current work is presented based on [2], [3], [4], [9], [10], [11], [12]. The core of KAOS consists of the following three activities [13]: hierarchical decomposition, goal-oriented reasoning, and obstacle management. The first activity deals with guiding through the refinement of goals all the way to requirements and operations. The second activity is related to reasoning about how a goal is satisfied from the satisfaction of decomposed sub-goals. The third activity is related to identifying things that may hinder the achievement of a goal and finding solutions to handle such situations.

KAOS was developed to provide a modeling framework to represent system specifications through different concepts related to the system objectives. These concepts include the refinement and operationalization of goals, considering constraints, together with the actions and objects that are responsible for and subjected to those constraints [4]. A model in KAOS can be one of the four basic models: goal, object, responsibility or operation. The four models are linked through external relationships with rules to ensure stability of the individual models and not to be altered by their inter-model relationships [14]. A model can be represented at three levels of abstraction: meta level, domain level and instance level. The models at the meta and domain levels consist of concepts (with attribute and relationships), relationships (with attribute and list of concepts) and attributes (consisting name, informal definition, domain of values and unit of values) [2]. For the purpose of our work, the goal model is the most used model at the domain level of conceptual abstraction.

Goals are important in the RE process because they drive the requirements elaboration. They give reason as to why the system is needed at the early phase of RE. A goal in KAOS is represented by a name, a natural language definition and a formal definition that is optional in a specification. A goal's *name* is preceded with a verb that states the goal *type* (depending on whether it is a behavioral or soft goal) [12]. Goal model diagram forms a hierarchy of goals through refinements from high-level to lower-level goals. The root of the tree represents the highest level goal, and the leaves represent requirements. A requirement is a goal that can be assigned to an agent as a responsibility. Reaching such a goal is an indication that the goal refinement should stop. Goal refinement can be either a top-down or a bottom-up process. When done top-down, lower-level goals are identified in order to achieve a high-level goal (answering the *how*). When done bottom-up, a lower-level goal is generalized into a high-level goal that it aims to achieve (answering the *why*). The general idea of this goal refinement process is that it must be possible to trace up the highest level goal from each requirement identified, as it is needed to refine each high-level goal to a set of requisites to ensure its satisfaction.

### 2.2 Natural Language Processing in Requirements Engineering

As the majority of sources for requirements are written in natural language, methods applying NLP and text mining techniques have received a lot of attention in the last two decades to help automate various RE tasks [15]. While the automation of the general RE process is too complex and beyond the current capabilities of such techniques [16], language engineering has been successful in addressing some problems in the early phases of RE such as RE elicitation and domain understanding [17].

Ontologies are a broad class of language engineering models, which are proposed as useful tools in RE [18], [19]. Ontologies define specifications and methods to formalize a set of concepts and their relationships [20]. Concepts are entities or mental abstractions described by natural language representations that are meaningful in a particular domain. For example, in the smart metering domain, there are concepts such as *smart meter*, *sustainability*, *demand response*, etc. In an ontology, concepts are linked to each other by a set of axiomatic relationships, forming directional, non-hierarchical graphs.

The benefits of the formalization introduced by ontologies have been recognized in various RE tasks. For example, ontologies can be used for different parts of a RE project to define knowledge, processes, domains, classifications and patterns [21]. These ontologies provide a tool to facilitate the enhancement and discovery of new requirements during the early elicitation phase [22]. They can be linked to the requirements specification documents and help manage requirements and trace sources, dependencies, and implementations of software [23].

Learning ontologies from text requires many intermediate NLP processing steps: from low-level tasks such as terms extraction, part-of-speech tagging, sentence parsing, to higher-level tasks such as topic identification, analysis of semantics, and relations discovery [24]. In this paper, we are interested in how to extract the entities of an ontology in form of goals and how to define the relationships among them. Learning goals have similarity with a relevant body of research that has been focused on abstraction identification, i.e., the extractions of concepts and entities of a domain [25].

Abstraction identification has been successfully applied during early RE phases to automate RE elicitation and domain understanding [17], [26].

The process of learning the structure of a general ontology is considered a difficult task and it is still an open problem in language engineering [7], [24], [27]. Learning taxonomies from text has been shown to be an easier problem to solve than learning ontologies. Taxonomies are a subclass of ontologies, i.e., lightweight ontologies [28]. Ontologies describe more general types of relationships, i.e., non-hierarchical and axiomatic relationships, and they are denoted as heavyweight or formal ontologies [29].

Different methods were proposed for taxonomy learning such as lexico-syntactic patterns [30], hyponymy information [31], noun-phrase matching [32], informatic-theoretic approaches [33], and graph-theoretic approaches [8]. The graph-theoretic approach is used in this paper. It is fast and it relies on few parameters in order to tune the overall structure of the taxonomy (the depth). It is based on general statistical relationships between taxonomy nodes, i.e., frequencies, rather than ad-hoc rules based on semantics and lexico-syntactic patterns. Semantics information is excluded from the taxonomy algorithm, but they are included in the goal extraction algorithm (node of the taxonomy).

Bibliometrics refers to the process of extracting useful information using text analysis from publication databases to identify science, technology, and innovation research and management opportunities [34], [35], [36]. Choosing the right source for the analysis depends on several factors that include quality of a database, nature of the domain of interests, desired depth of the investigation and costs [35]. For example, conference and journal papers often contain newer material than books and patents and as such often contain the most recent trends and technologies. Books and patents require a longer publishing process and their contents include more specific, deeper material. Novel material can appear in different journals from different subfields. For RICS, due to their nature, academic papers might provide an optimal level of abstraction, and as such, they are chosen as suitable source for the case study.

## 3 Research Method

In the preliminary step of this study, we performed a case study with qualitative evaluation [37]. A case study allows exploration and understanding of complex issues by providing a holistic, in-depth investigation of system behavior [38]. The objective of the preliminary step was to evaluate how effective KAOS is to guide the specification of goals for a RICS based on information acquired from research publications. Guiding goal specification for a system requires identifying relevant goals that system should achieve. Therefore, KAOS was evaluated on how useful it was to extract relevant goals for RICS.

We used the smart metering system as the unit of analysis to evaluate the findings during the exploration and understanding of the system's goals and requirements. We evaluated KAOS on its appropriateness to elicit goals from research publications and based on how reliable and useful the identified goals were with respect to real systems. We compared the identified goals with real system goals (reverse engineered from the implemented early devices) to check for their relevance, i.e., if they represent goals that are satisfied in the implemented early models.

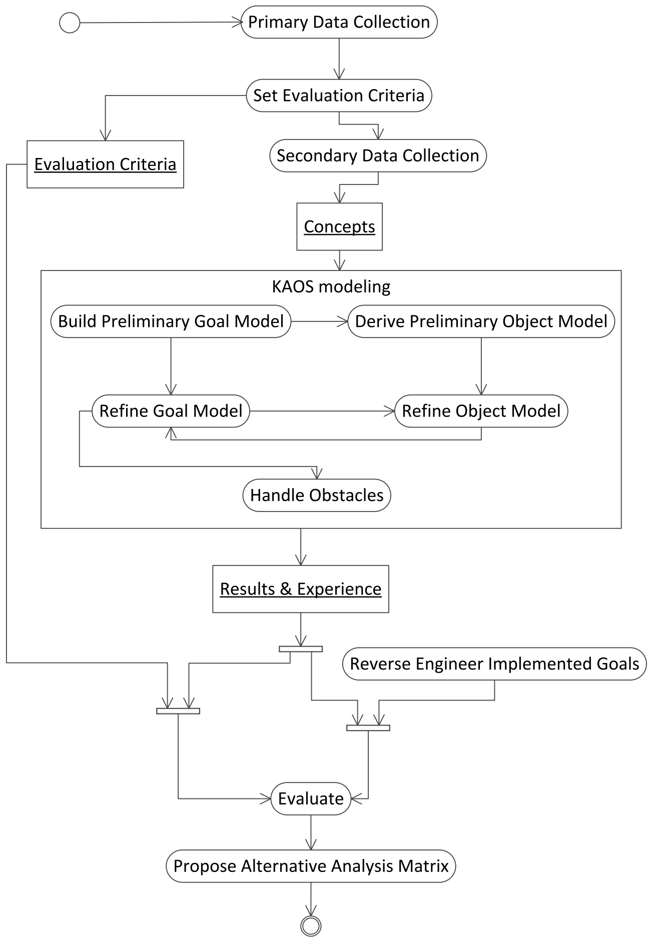


Fig. 1. Preliminary research process.

To serve its purpose, the preliminary research step provided answers to the following research question: *How reliable and useful are the goal models obtained by applying KAOS on RICS compared to the implemented models?*

The tasks performed in the preliminary step are shown in Fig. 1.

The case study execution included three major steps: preliminary data collection, application of KAOS on the smart metering system, and the evaluation of the results.

Data collection in qualitative research method can be one of three different forms [38]: in-depth responses to open-ended interview questions, rich and detailed descriptions of direct observations or extracts from written publications keeping the context intact. In line with our objective of extracting goals from research publications, we use the third method to collect our data. Strategies provided by KAOS, taking advantage of the language constructs, are applied during data collection to keep the context intact.

The KAOS process steps were followed as the knowledge acquisition and model building steps provided in [12].

*Step 1: Build preliminary goal model.* The first step in the process of RE using KAOS is investigation and analysis of available sources to identify stable goals of the existing system. Identified goals are represented by a set of attributes, a name and its definition, which are mandatory.

Other optional attributes of a goal are the goal type (if it is a softgoal or behavioral), the goal category (functional or non functional), the source, priority, issue and formal specification.

*Step 2: Derive preliminary object model*. Objects are identified from the goal definitions in Step 1, or system descriptions encountered in the documents. Stable concepts referred to in the goal definitions or domain constants which represent an entity, association, attribute or event are looked for when identifying objects.

*Step 3: Updating the goal model through WHY and HOW questions*. This step is similar to Step 1, but concentrates on goals that particulary represent the system-to-be. It includes identification of alternative ways to achieve the stable goals identified in Step 1. Alternative responsibility assignments are also made by identifying alternative agents. Higher-level goals that may have emerged because of new technological development, opportunities or threats are also identified. The alternative sub-goal refinements and agent assignments must meet the higher-level goals, both the stable ones and the new ones. New goals are identified from the documents by looking for intentional word search similar to Step 1.

*Step 4: Updating the object model*. This step is similar to Step 2. Based on the modified goal model, the object model is also modified to include newly identified objects or remove objects that are no more needed.

*Step 5: Analyze obstacles, threats and conflicts*. In this step, the exceptional events that may hinder goal satisfaction are looked for. Negating the low-level goals can be refined into all possible reasons as to why a goal achievement may fail. Threats are when agents fail to achieve the goals they are responsible for, or in other words, when they malfunction to achieve their anti-goal. Conflicts between goals are also looked for at this stage.

The results of the preliminary step are presented in [37] and they have provided the context for the execution and evaluation of the NLP-KAOS method described in Section 4, which can be classified as design research study [38], [39].

## 4 Data Acquisition and Natural Language Processing

The NLP-KAOS approach aims to support requirements engineers in mining goals and discovering their relationships from text data. The scheme in Fig. 2 summarizes the computational steps of NLP-KAOS: data collection, goal extraction and goal modelling. Fig. 2 depicts also the necessary points of interaction between the analyst and the computational method. The overall data mining process is a semi-automatic and iterative process, as the analyst is required to iterate in order to refine the information at each stage, e.g., collect new data, tune parameters of mining algorithms, etc. Through this process, the information from the list of goals and from the taxonomy can support the requirement engineer to build object models using KAOS.

### 4.1 Bibliometric Data Acquisition

NLP-KAOS requires as an input a collection of research publications related to the RICS under investigation. Research publications can be collected from professionally

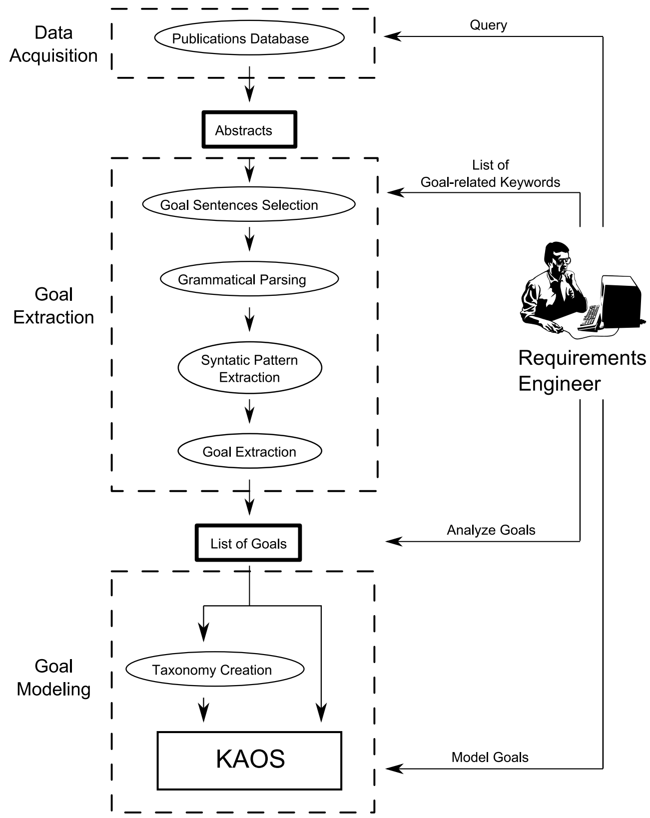


Fig. 2. NLP-KAOS method.

manually curated online repositories such as SCOPUS, Google Scholar, web of science, etc. In general, these repositories do not allow automatic querying and the access to textual data is limited to public. Thus, as a first step, it is necessary to query the front-ends of these repositories with appropriate keywords in order to retrieve the relevant publications. As per [35], different querying techniques can be used depending on how narrow, general or sparse the topic of interest is. Next, the abstracts of each retrieved publication are downloaded for the further processing stages.

### 4.2 Goal Extraction

The aim of this step is to provide the requirements engineer with a preliminary lists of goals extracted from the abstracts. The process of goal extraction is divided in four parts: goal-related sentence selection, grammatical parsing, rule-based syntactic pattern extraction and goal-keywords identification.

#### 4.2.1 *Goal-Related Sentence Selection*

In order to minimize the computational demand on the following processing steps, only sentences which contain goals should be analyzed. As suggested in [40], a list of goal-specific keywords defined by the analyst is compiled to isolate only goal-related sentences. Table 1 shows the list of intentional and amelioration keywords, some of them suggested in [40], used in this paper during the analysis of the smart metering system case study. In particular, keywords are either verbs or nouns (or both, e.g., 'aim'). This list is further expanded to include various verb forms, e.g., 'aim', 'aims', 'aiming', etc.

TABLE 1
List of Goal-Related Keywords

| | |
|---|---|
| Intentional | objective, aim, purpose, achieve, maintain, avoid, ensure, guarantee, want, wish, motivate |
| Amelioration | improve, increase, decrease, reduce, enhance, enable, support, provide, make |

#### 4.2.2 Grammatical Parsing

The grammatical structure of each goal-related sentence and the lexical categories of its constituent words are extracted using the Stanford Parser [REF]. The Stanford Parser is an open source Java implementation of a probabilistic natural language parser trained on the Wall Street Journal document collection. In this paper case study, version 2.0.5 of the parser is used. The Stanford Parser generates syntax trees from sentences, as shown in the example in Fig. 3 for the phrase: *A smart meter is an electrical device that records consumption of electrical energy and enables the measurement of energy in both directions.* This sentence was identified in the previous step as it contains the goal-related keyword *enables*. Fig. 3 shows the tags of various part of the example sentence positioned as inner nodes of the tree, while single words of the sentence are always the leaves of the grammar tree.

#### 4.2.3 Rule-Based Syntactic Pattern Extraction

The core part of the goal extraction algorithm is to identify the recurrent syntactic patterns in the parsing trees, which can be transformed as goals. As natural language is not strict, precise and sometimes not correct (biased by mistakes and grammatical errors), extracting goals may require a large numbers of ad-hoc extraction patterns. Goals may be hidden in the natural language, which any NLP techniques and rules may find hard to identify. This paper approaches these problems from a statistical perspective: *using simple common patterns and applying to a the large amount of textual data, while some goals will be lost, the most relevant goals will appear often in the data.* Thus, a general and simple pattern of interest for goals is considered: the triplet <Predicate-Object-Prepositional Phrase>. Referred to as a goal, the predicate (the verb) conveys the action taken upon the entity described by the object. The prepositional phrase is also considered because it often contains some important information about the action or it provides some important attributes for the object. For instance, in Fig. 3 two triplets of interest may be extracted: *<records, consumption, of electrical energy>* and *<enables, the measurement of energy, in both directions>*.

Two simplifications were considered in this version of the proposed approach. First, the subject related to a triplet is not included. The use of the triplet <Predicate-Object-Prepositional Phrase> is assumed to be sufficient, as the purpose of the analysis in this paper is exploratory. High level goals may be understood from the context and they often refer to the main system, e.g., smart meter. Furthermore, the subject of a triplet is difficult to extract, as it may not be explicit in the sentence: defined in a different sentence before or be embedded in personal pronouns, which would require other NLP tools such as anaphora resolution. Second, the approach in this paper does not extract all possible <Predicate-Object-Prepositional Phrase> in the text, but only the ones related by goal-related keywords. Thus, in the example in Fig. 3 the triplet <enables, the measurement of energy, in both directions> is discovered, but <records, consumption, of electrical energy> not, even if it can be classified as a valid goal for a smart meter.

We developed a pattern-based grammatical algorithm to extract the goal-related triplets from the Stanford Parser syntactic trees. This ruled-based grammatical algorithm is inspired by the algorithm presented in [41], which was implemented to extract <Subject-Predicate-Object> triplets from text as part of a methodology that generates summaries of documents [42]. Fig. 4) shows the pseudocode of the four main subroutines of the algorithm used in this paper to extract goals from syntax trees.

The main function of the algorithm takes as input the parsed sentence and the goal-related *keyword* of interest. If *keyword* is a verb such as 'avoid', the algorithm identifies the highest containing subtree labeled with the *predicate tag VP*, *treeVP*. The latter tree is passed to the function ObjectExtraction that returns both *object* and prepositional phrase *PP*. if *keyword* is a noun such as 'purpose', the algorithm identifies its highest containing subtree labeled with the *noun tag NP*, *treeNP*. The function looks at the rightmost VP sibling of the tree *treeNP* which is then passed to PredicateExtraction and ObjectExtraction functions.

ObjectExtraction receives the *treeVP* subtree as input. The object is found by seeking in the VP subtree for the deepest noun, i.e., words with NN, NNP, NNPS or NNS tags, or by seeking an adjective in the ADJP tree, i.e., JJ,

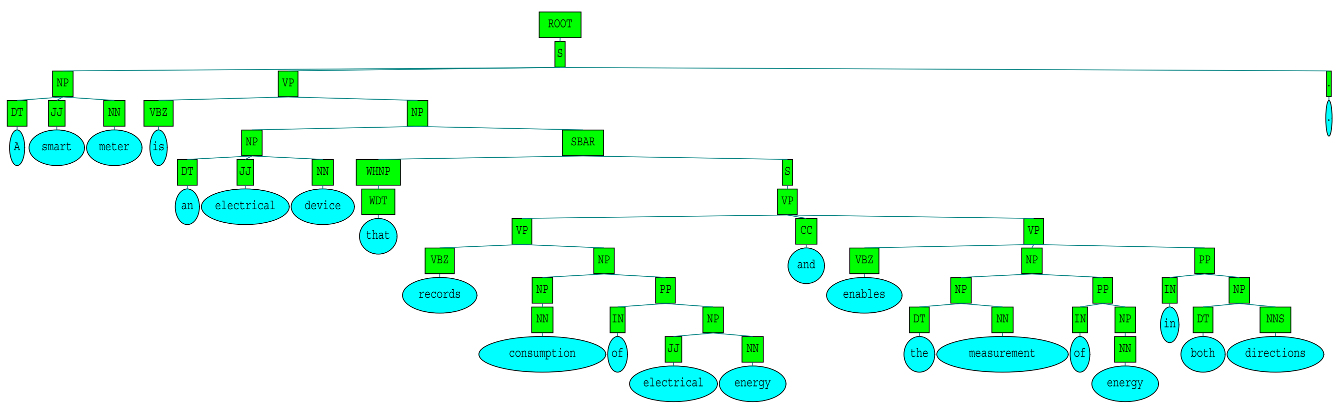


Fig. 3. Parsing of an example sentence.

```
Require: highestVP(keyword) and highestNP(keyword) compute the highest VP or NP subtree
    containing the keyword.
Require: sibling(tree) compute the siblings trees of tree
Require: label(tree) the main tag of tree
Require: leaves(tree) return the terminal nodes of tree

function MAINGOALEXTRACTION(parsedSentence, keyword)
    if isVerb(keyword) then
        predicate ← keyword
        treeVP ← highestVP(keyword)
        object, PP ← OBJECTEXTRACTION(treeVP)
    end if
    if isNoun(keyword) then
        treeNP ← highestNP(keyword)
        treeVP ← sibiling(treeNP) if label(treeVP) is VP
        predicate ← PREDICATEEXTRACTION(treeVP)
        object, PP ← OBJECTEXTRACTION(treeVP)
    end if
    return predicate, object, PP
end function

function OBJECTEXTRACTION(treeVP)
    subtrees ← sibiling(treeNP) if label(subtrees) is NP,PP,or ADJP
    for tree in subtrees do
        if label(tree) is NP or PP then
            object ← leaves(tree) if label(subtrees) in [NN, NNP,NNPS or NNS]
        end if
        if label(tree) is ADJP then
            object ← leaves(tree) if label(subtrees) in [JJ,JJR or JJS]
        end if
    end for
    PP ← PREPOSITIONALPHRASEEXTRACTION(object)
    return object, PP
end function

function PREDICATEEXTRACTION(treeVP)
    for tree in treeVP do
        if label(tree) is not S then
            verb ← leaves(tree) if label(subtrees) in [VB,VBD,VBG,VBN,VBP,VBZ,MD]
        end if
    end for
    return verb
end function

function PREPOSITIONALPHRASEEXTRACTION(keyword)
    treeNP ← highestNP(keyword)
    subtrees ← in sibiling(treeNP) if label(subtrees) is PP
    PP ← in leaves(tree) if label(subtrees) is not PP
    return PP
end function
```

Fig. 4. Pseudocode of the rule-based extraction algorithm for the Triplet <Predicate-Object-Prepositional Phrase>.

JJR or JJS. Before returning the *object*, the function ObjectExtraction obtains *PP* from PredicateExtraction function. Both *object* and *PP* can be empty strings if no results are found.

PredicateExtraction identifies the deepest verb in the VP input tree, as it is usually associated with the main action. In particular, it seeks words in the *treeVP* children which are tagged as [VB,VBD,VBG,VBN,VBP,VBZ]. However, the loop in PredicateExtraction avoids entering in subordinate clauses S or SUB, e.g., 'which...', as they usually refer to different semantic part of a sentence.

The function PrepositionalPhraseExtraction takes all the PP siblings tree of the VP subtree, and returns all tagged words belonging to the PP tree except of the articles, e.g., DT. The function PrepositionalPhraseExtraction also checks if there is the preposition 'of' label as IN in the text. In this case, it returns the *PP* with '[OF]' instead of 'of'. The latter part of the algorithm is explained in the following sections in more detail.

#### *4.2.4 Goal-Keywords Identification*

The final part of the algorithm takes all the <Verb-Object-Prepositional Phrase> triplets and produces a list of *Verb [Goal-Keywords]*. Goal-Keywords are formed in two phases: by keeping the nouns of the Object in the triplet; by including the nouns of the Prepositional Phrase if the preposition is 'of'. Thus, in the example in Fig. 3, the triplet <enables, the measurement, of energy in both directions> is transformed to ENABLES[[OF] energy measurement]. The particle '[OF]' is kept to distinguish the origin of Goal-Keywords.

All Goal-Keywords retrieved are searched for again in the abstracts. During the search, the Goal-Keywords are stemmed to take into account identical morphological roots. The number of Goal-Keywords occurrences in each abstract is saved in a matrix. Statistics of the number of occurrences is useful for the requirements engineer to identify the most relevant goals. Moreover, these statistics are important in order to construct taxonomies as explain in the next section.

### 4.3 Taxonomy Creation

A taxonomy simulates the refinement process required during development of goal models from high-level to lower-level goals and requirements. We used a semi-automatic method for taxonomy generation based on the Heymann algorithm for this purpose. Originally, the Heymann algorithm was used to create navigable hierarchical structures of collaborative tagging systems [8]. In [43], [44], the algorithm was modified to be used for bibliometric data. To use the Heymann algorithm for goal modeling, the following tasks need to be addressed:

1) define the relationships between goals, and
2) extract the hierarchical structure from the relationships between goals.

The process of eliciting goal models in KAOS consists of iterative process of goals refinement by asking *how* and *why* questions. This is often a challenging task, which requires the subjective input from the analyst, and which is difficult to reproduce using a concrete algorithm. Instead, our approach is based on a correlation analysis, which is commonly used in text mining. The correlation analysis is based on the distributional hypothesis which defines that two words that appear in many similar linguistic contexts are also semantically related [45]. As such, goals that appear in the same document are considered to be linked. The refinement from high-level to lower-level goals is achieved by inspecting the distribution of goals in all corpus. Thus, the first step of Heymann algorithm is to compute a matrix of statistical correlations between goals, which is denoted as *similarity matrix*.

The similarity matrix is constructed by computing the statistical correlation among Goal-Keywords. To construct the similarity matrix, each Goal-Keyword $u$ is searched in the abstract database in order to build the vector $\mathbf{x}_u = [x_1, \ldots, x_D]$ of length equal to the number of documents in $D$. Each $\mathbf{x}_u$ is a binary-term vector: $x_i$ is $1$ or $0$ depending on weather the Goal-Keyword $u$ is present or not in the $i$th abstract. The relationships between Goal-Keywords $u$ and $v$ is computed by the standard cosine vector similarity defined as:

$$S_{cos}(u, v) = \frac{\mathbf{x}_u \dot{\mathbf{x}}_v}{\|\mathbf{x}_u\| \|\dot{\mathbf{x}}_v\|.} \tag{1}$$

Thus, a threshold parameter $\tau_s$ is used to set to $0$ relationships with low value in the similarity matrix. $\tau_s$ prevents spurious correlations that might have occurred by chance. The modified similarity matrix can be transformed into a weighted similarity graph $\mathcal{G}_S(V, E)$, with the nodes $V$ representing goals and the edges $E$ representing the non-zero similarities. An example of graph is shown in Fig. 5 [44].

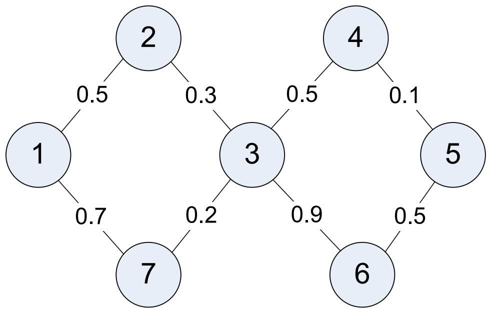


Fig. 5. Example graph from similarity matrix.

The core part of the Heymann algorithm takes the graph $\mathcal{G}_S$ and inserts each of its goals into a new hierarchical $\mathcal{T}_S(V, E_1)$ structure, i.e., the taxonomy. First, each goal is ranked in $\mathcal{G}_S$ by a measure of generality. The generality of a goal can be defined by how it occurs with its lower level goals. The greater (smaller) the generality, the more broad (specific) is the goal, the more (less) connected is its node in $\mathcal{G}_S$, the higher (lower) it sits on the taxonomy. From network theory, the concept of *centrality* is appropriate to measure the generality of as node. In practice, centrality measures how well connected is a node in the graph. Common centrality measures considered in the literature are:

$$c_B(v) = \sum_{v \in K \setminus s, ts \neq t} \frac{\sigma_{st}(v)}{\sigma_{st}}, \tag{2}$$

$$c_C(v) = \frac{1}{\sum_{t \in K \setminus v} (1 - NGD(v, t))}, \tag{3}$$

where $c_B(v)$ is the *betweenness centrality*, while $c_C(v)$ is the *closeness centrality*. With respect to a simple measure of degree centrality (number of links incident upon a node), which is local, both betweenness centrality and closeness centrality are global metrics. They are based on the shortest path among not adjacent nodes and they are more reliable on the global structure of the graph. In Fig. 5, node 3 has the largest centrality, followed by nodes $2, 4, 6, 7$ and by nodes $1, 5$. The betwenness and closeness are the same for this example.

Second, each goal is inserted into a growing taxonomy structure in the order of decreasing centrality. Starting from the root, goals are either added to the root or to the most similar node already present in the taxonomy. The threshold parameter $\delta_C$, denoted as the *child penalty*, is used to control the latter decision. If the similarity between goals is greater than $\delta_C$ the keyword is attached to the most similar node, otherwise it is attached to the root. The child penalty threshold provides a control of the depth of the taxonomy, i.e., the higher penalty $\delta_C$ creates a deeper taxonomy. In Fig. 5, depending on $\delta_C$, node 1 is either connected to the most similar node 7 or to the root.

A further modification that is used in this paper Heymann algorithm is the concept of *iterative centrality re-ranking* [44]. After each insertion of a goal into the taxonomy, the algorithm eliminates the goal from $\mathcal{G}_S(V, E)$ and recomputes the centrality of the remaining goals. Iterative centrality re-ranking takes in consideration the case of word with similar meaning, i.e. it possible that some authors prefers to use one term instead of other ones. For example, the terms "AMI" and "Advanced Metering Infrastructure" or "power network" and "power grid" are both used, but authors usually settled for one of them. In the example in Fig. 5, both 1 and 5 may be "super-topics" for $2, 7$ or $4, 6$. However, in the original ranking they have the least centrality. After removing the node 3, if centrality is recomputed, both 1 and 5 jump on top of the ranking. The disadvantage of re-ranking is the higher computational speed due to the re-calculation of the centrality, which increases of a factor of $O(n)$ the overall algorihtm.

## 5 Smart Metering System Case Study

The smart metering system is a core subsystem of a smart grid [46]. Smart grid is a term used to describe the next class of power systems that aims to improve and make more efficient the electrical power generation, transmission and distribution. The smart grid is an example of RICS: a modern, large, and complex sustainable system, which extensively uses software services on top of the power infrastructure to improve functionality, user experience, security, etc. [47]. Smart grid aims to enhance the security and reliability of the power infrastructure, to operate resiliently against physical and cyber attacks, to be self-healing from power disturbance events, and to provide power quality for future energy supply needs [48], [49].

A smart meter is a metering device that enables the automatic collection of information about the energy usage in the grid. Importantly, it uses a two-way communication system that establishes an interaction between the utility companies and the consumers. The communication in both ways allows the consumers to have more control of their energy usage, helps the utility companies cut manual metering costs, and provides better monitoring of the electrical systems. Smart meters enable the active participation of consumers in energy saving efforts.

### 5.1 Natural Language Processing Results

The SCOPUS database was queried to identify the most relevant academic publications on smart metering systems. The front-end of the databases was queried using the search key: ("smart meter" OR "smart metering" OR "AMI" OR "Advanced Metering Infrastructure") AND ("electric" OR "grid"). The result of the query returned a total of 1,569 records. The abstracts of these records were downloaded and stored in a local SQL database. This consisted of 540 journal articles, 978 conference papers and 51 other publication (reviews, editorial, etc.).

The first stage of the proposed approach aims to extract relevant goals in the form of a list, to be submitted to the requirements engineer for an initial stage of exploratory analysis. In particular, as discussed in Section 4.2, two lists are of interest during the process of goal elicitation: the full list of <Predicate-Object-Prepositional Phrase> and the list of proposed goals in the form of Verb[Goal-Keyword].

The first list presents some examples of <Predicate-Object-Prepositional Phrase> extracted from text (their location in the database may be included for a further reference).

TABLE 2
Examples of <Predicate-Object-Prepositional Phrase> Triplets

| | | |
|---|---|---|
| increase | energy efficiency | - |
| improving | energy efficiency | - |
| improve | electricity billing | - |
| enabling | decisions | - |
| purpouseOFobtaining | information | |
| enables | demand management | in homes |
| increases | cost | [OF] power supply |
| achieve | energy efficiency | via two way interaction |
| enabling | SG | [PRED] to react possible impact [OF] malicious activity |

A total of 647 triplets where extracted from the database. Some representative examples of this list are shown in Table 2. The first four entries of Table 2 shows that the extracted triplet do not have the corresponding prepositional phrase. Some of them are self sufficient with the prepositional. However, <enabling, decisions,-> is vague as it is. The fifth triplet shows the case of a noun word 'purpose' which has been associated a predicate 'obtaining'. The other entries of the table shows full triplets included of the prepositional phrase. In particular, the algorithm identified [OF] phrases where Goal-Keywords can be identified, e.g., power supply cost. The algorithm also warns the engineer about predicate subclauses with [PRED], as in the last example.

The second list, as shown in Table 3, presents some examples of proposed goals in short form. The list is ranked by the number of occurrences of the Goal-Keywords found in the abstracts. Goal-Keywords are extracted from the previous list by selecting the unique set of Object elements (stripped by articles) and enriched, if possible, by keywords from the Prepositional Phrase part of the triplets. The list is divided in two parts to show, as an example, the first 10 most frequent goals as well as some less frequent significant goals extracted by the algorithm. We have shown them together in order to emphasize that there are important goals among those that appear less frequently, i.e., frequency of goals should not be the only criteria used in filtering goals, and the full attention should be paid to all the extracted goals. The only judgement that we can correlate to the frequency of goals is that they tend to be higher-level, more general goals as compared to those occurring less frequently.

The second stage of the approach is to assemble the most relevant Verb[Goal-Keyword] into a taxonomy. The requirements engineer can tune the structure of this taxonomy by varying the internal parameters of the Heymann algorithm. A final version of the structure of this taxonomy is shown in Fig. 6. As the full taxonomy is too large to be visualized in detail, only a subset of representative nodes were highlighted and displayed in Fig. 6. The most important parameter for the tuning of the taxonomy was the child penalty $\delta_C$. This parameter was varied to control the level of depth of the taxonomy. A low value of $\delta_C$ is consistent with a flat taxonomy: all keywords would be attached only to the central node. A flat structure does not give any useful information to the requirements engineer and it must be avoided. As a rule of thumb, during the construction of the taxonomy, the requirements engineer selected the minimum $\delta_C$ that left no leaves nodes attached to the root as shown in Fig. 6.

## 6 Case Study Evaluation and Goal Modeling

We have evaluated the results of NLP-KAOS in two steps. First, to reason about the validity of the NLP-KAOS results, we have compared the list of NLP-KAOS elicited goals with a list of manually, independently, elicited goals. These manually elicited goals were evaluated and validated with respect to the goals implemented in a number of industry-developed smart metering devices [37]. Second, we have derived partial goal models for the system to demonstrate how the three forms of the goal results provided are useful for goal modeling. This way, the results of this paper were evaluated with respect to: manually elicited goals from research publications; with independently implemented goals in retail smart metering devices; and usefulness of the NLP-KAOS elicited goals for the goal modeling.

The independent manual goal identification was performed similar to the automated analysis by first collecting documents from online sources with a use of a search keywords to enhance the relevance of results. The goal model parts were built based on the goals obtained, according to the steps outlined in [3], [12]. Basically the list is the source of the goals and the taxonomy is used to refer on how goals are related to each other. Deriving goal models starts with selecting a goal from the list of goals in the form verb[Goal-Keyword]. The *verb* is mapped to one of the goal types ACHIEVE, MAINTAIN, AVOID, CEASE, IMPROVE, INCREASE or DECREASE [12]. The goal is then found in the taxonomy in which its links with other goals that are studied for potential relationships that may aid in goal refinement. These relations could be contributions of one goal to another (relating high-level goals to lower-level goals), the different aspects of a particular goal, or problems or obstacles and potential solutions.

The manual goal extraction was based on 44 documents collected from the SCOPUS data set. The search keyword *"smart meter" AND (electric OR "smart grid") AND (goal OR interest OR derive OR objective OR aim OR purpose OR require OR intent OR concern OR "in order to")* was used on the titles and abstracts of documents. Each conjunctive addition to the list of keywords reduced the number of documents retrieved from 1,456 to 870 and then to 64. The obtained documents were filtered to 44 through analysis of the titles and elimination of the documents not related to smart metering system. A list of 41 high-level goals was prepared based on the abstracts and introductions of the documents.

TABLE 3
Proposed Goal List (Verb[Goal-Keyword])

| Verb | Goal-Keyword | Freq. |
|---|---|---|
| FIND | [CTs] | 3729 |
| MAINTAINING | [meters] | 2378 |
| ENABLE | [smart meter] | 2347 |
| IMPROVE | [grid] | 2199 |
| IMPROVE ENABLE SUPPORT ENABLE | [smart grid] | 1975 |
| IMPROVE AIMofDELIVERING | [electricity] | 1457 |
| SUPPORTING | [AMI] | 1170 |
| ENABLING | [communication] | 1024 |
| ENABLES | [management] | 728 |
| ENABLES | [[OF] smart meters use] | 709 |
| IMPROVE | [energy saving] | 254 |
| PURPOSEofCAPTURING | [energy consumption data] | 166 |
| ENABLES | [[OF] energy measurement] | 143 |
| SUPPORT | [demand-side management] | 100 |
| IMPROVE | [customer service] | 99 |
| IMPROVE | [electricity billing] | 87 |
| MAINTAINING | [information privacy] | 71 |
| ACHIEVES | [low power consumption] | 70 |
| ACHIEVE | [load balance] | 53 |
| IMPROVE | [communication quality] | 51 |
| ACHIEVE | [automation services] | 47 |
| ENABLES | [bi-directional communication] | 30 |
| IMPROVE | [meter data quality] | 30 |
| MAINTAIN | [[OF] electric power quality] | 30 |
| ENABLING | [[OF] data exchange] | 26 |
| SUPPORTING | [[OF] home appliances monitoring] | 22 |
| ENSURES | [resiliency] | 22 |
| IMPROVING | [[OF] measurement accuracy] | 20 |
| IMPROVE | [supply security] | 20 |
| SUPPORT | [power restoration] | 17 |
| IMPROVING | [electric system stability] | 17 |
| ENABLES | [time-of-use tariffs] | 17 |
| INCREASES | [off-peak demand] | 15 |
| ENSURING | [frequency regulation] | 15 |
| ENSURE | [reliable energy supply] | 14 |
| ENABLES | [higher efficiency] | 12 |
| ENSURE | [[OF] smart meter network efficient operation] | 12 |
| IMPROVE | [[OF] smart grid systems efficiency and reliability] | 12 |
| AVOID | [potential outage] | 11 |
| INCREASING | [[OF] smart grid security level] | 11 |

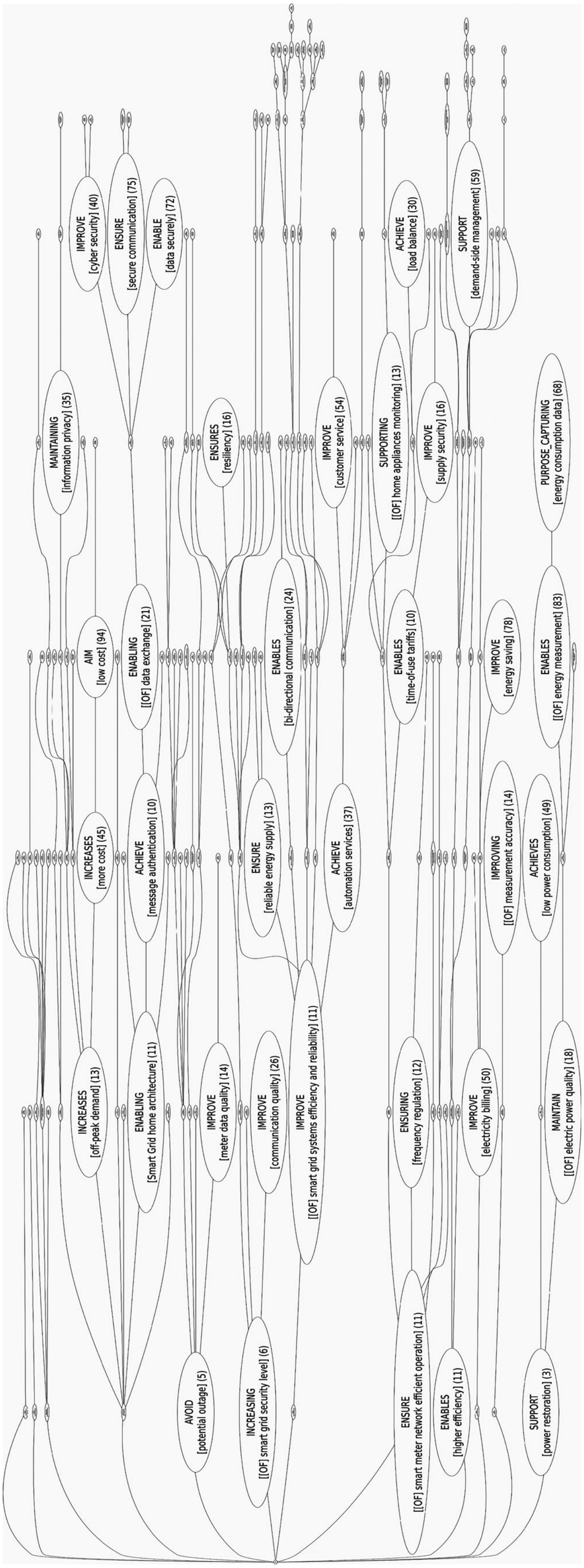


Fig. 6. Generated taxonomy.

This list of goals was evaluated and validated in comparison to the goals implemented in a selection of concrete smart metering devices [37].

TABLE 4
Manual versus NLP-KAOS Elicited Goals

| Manually Elicited Goals | NLP-KAOS |
|---|---|
| better demand management | ✓ |
| improved settlement accuracy | ✓ |
| platform to measure and aggregate electricity export from microgenerators | ✘ |
| improved theft detection | ✓ |
| reduction of network losses | ✘ |
| improved fault management | ✘ |
| load management and forecasting | ✓ |
| enhanced investment planning | ✓ |
| enhanced network management | ✓ |
| reduce peak demand of energy | ✓ |
| support time-of-use concept of billing | ✓ |
| reduce cost of meter reading | ✘ |
| lower energy source generation cost | ✓ |
| elimination of inaccurate and estimated bills | ✓ |
| more ability to manage energy use | ✓ |
| wide choice of new tariffs and products | ✘ |
| flexible payment options | ✘ |
| easier business with energy company | ✘ |
| lower electricity cost | ✓ |
| power saving | ✓ |
| make informed decisions | ✓ |
| improved customer service | ✓ |
| reduction in cost to serve | ✓ |
| improved debt prevention capability | ✘ |
| extended range of products and services | ✓ |
| platform to deliver carbon reduction targets | ✘ |
| simplification of complex electricity market processes | ✓ |
| profitability | ✘ |
| securing energy and related carbon saving | ✓ |
| improve energy market function and supply competition | ✓ |
| renewable target | ✓ |
| developing markets for energy services | ✓ |
| zero carbon homes | ✘ |
| sustainability | ✓ |
| energy efficiency | ✓ |
| energy conservation | ✓ |
| automatic measurement | ✓ |
| improve power quality | ✓ |
| achieve power saving | ✓ |
| home automation | ✓ |
| demand response | ✓ |

### 6.1 Comparing Automatic Results with Manual Goal Elicitation

From the list of 647 triplets, a set of 461 goals are extracted. The list can be reduced by setting a threshold on the number of occurrences, in order to avoid unique or not relevant Goal-Keywords. If unique occurrences are eliminated the set of goals becomes 343. The list can be reduced further (with five or 10 keywords then the goals step down to 244 or 201). However, the list of 343 is considered for comparison with the list of 41 high-level goals that was manually elicited from the documents [37]. As shown in Table 3, lower ranking goals may be still informative to the requirement engineer. The result of the comparison is shown in Table 4. There are approximately 75 percent of the goals from the manual list that were also identified by the automated method. Even with the larger number of goals extracted using the automated method, there are some goals missing.

Differences between the automated and manual results are attributed to a number of reasons.

*Goals elicited using the automated method but not using the manual method:* More goals were identified using the automated method because:

- The number of documents used as source of goals for the automated method (1,569) is greater than the number of documents used to extract goals manually (44). A greater number of documents might imply a wider coverage of the topic at hand and thus resulting in a higher number of goals.
- Manual goal identification did not completely rely on the helpful keywords (intentional, amelioration, etc.), although they were used to facilitate the process. Analyst's judgment was used at the end to decide whether a goal is valid or not. However, the automated method relies completely on keywords and not analyst's judgment. The automated method identifies and presents all sentences with potential to contain goals if they happen to have the same verbs and structure as defined by the algorithm. Therefore some goals that are not be relevant within the context may also be included, e.g., the elicited goal IMPROVE [water management].
- While doing the goal identification manually, a few goals might have been missed due to human error. On the other hand, the automatic method does not miss any statements that have goals as long as they have the structure or keywords predefined by the algorithm.

*Goals that the automated method missed.* It is recommended to make use of keywords, such as intentional, amelioration, problem, etc., during goal elicitation. This works best if the documents used as a source of goals are actual problem statements and requirements (or more generally needs and wishes) of stakeholders. In those cases, more prescriptive statements would be found in the documents. However, in the case of our study, academic publications were used as the source of goals. Since these documents are not intended for the purpose of requirements or goals identification, the general expected form of most of the statements is descriptive form. This means that many of the intentional keywords may not be available in the documents. The list of keywords was prepared to cover the words that are most likely to be found and be helpful. However it is not an exhaustive list of words that may express intention or goal.

*Differences in the type of goals extracted.* The results suggest that the manual goal elicitation process was focused more on the high-level goals that different stakeholders would like to achieve using the smart metering system. Therefore the analyst concentrated on the diversity of goals that could be identified to be important regardless of the solution to be used. The goals where then refined using *why* and *how* questions. Then, goals related to using the smart metering system were further refined down to the basic functionality. On the other hand, the automated method was not performed by using any judgment of the analyst, and it identified as many goals as possible (regardless of their level in the goal hierarchy). Therefore, based on the nature of most of the academic publications, which intend to provide a solution for some particular problem, most of the goals identified were focused towards system goals rather than stakeholder goals.

In addition, more of the non-functional category [12] goals were identified at their meta and domain-levels such as *efficiency, network efficiency and electricity efficiency; security, data security, network security and electricity supply security; privacy, data privacy and user privacy*.

The results of the automated include more specific goals. For example goals identified to concern the utility in general, in the manual identification were identified as specific goals of transmission, distribution and generation. Some example goals are IMPROVE[power system reliability], MAINTAIN [distribution grid reliability], IMPROVE [transmission grid] and ENABLES [efficient energy generation].

Finally, using quantitative analysis of the automated method, it is possible to define standard information retrieval metrics such as precision and recall:

$$Precision = \frac{\|A \cap M\|}{\|A\|} \cong 0.1, \tag{4}$$

and

$$Recall = \frac{\|A \cap M\|}{\|M\|} \cong 0.7, \tag{5}$$

where $A$ is the set of automatically extracted goals and $M$ is the set of manually extracted goals. The precision metric is computed by setting the denominator $\|A\| = 343$ in Equation (4). The factor $\|M\|$ in the denominator of Equation (5) is based on the assumption that the manual method is considered as a golden reference.

The low value of the precision is due to the greater number of goals extracted by the automated method. This may not be a crucial problem depending on the focus of the method. It may be useful to have a large number of goals detected in a relatively cost effective way that can be used for more detailed goal analysis using traditional methods. For example, if the analysis is done to produce a reference goal model for the general reference architecture, then the analyst will be probably primarily interested in most frequently occurring goals, and if the analysis is done for finding some obscure goals that might provide competitive market advantage then the least frequent goals might be of the highest interest. For the study of the system evolution or strategic planning, most likely the most interesting are the goals of the average frequency. Each use case creates its own challenges and cutoff points that should be subjects of the future study. In any case, we found that the cost of analyzing the generated list is much lower than the cost of the manual goal mining. For the most straightforward use case of using the results of the automatic goal elicitation for the actual product development (as a secondary source of goals), the recommended practice should be to go over each proposed goal with actual stakeholders in order to agree on the optimal list of goals. In such a case, the automatically generated list will serve at least as a source of goals that can help ensure comprehensiveness and minimize the likelihood of missing an important goal by a stakeholder.

A possible way to increase precision is to select the top most relevant goals as suggested in [26]. However, it is not possible to use the ranking by frequency as discussed in Table 3, i.e., the relevant goals may be found at lower frequency ranking. At the moment, the analyst's judgement must be included in goal selection and their prioritization and shortlisting.

### 6.2 Deriving Goal Model

The list of automatically elicited goals provides us with almost ready to use goals to build goal models. The required step is to map the verb of a goal that should be included in the model (verbs such as ENABLE, SUPPORT, ENSURE, AIM) to one of the KAOS goal types listed earlier in this section. This is an important step that is constrained by the KAOS language constructs and constraints. In order for this to be done, we must perform it manually as we are not able to infer the context and semantics of each goal in the automated fashion. Examples of such goals are (where "→" means "changed to")

- ENABLES [time-of-use tariffs] → ACHIEVE [time-of-use tariffs]
- ENABLING [smart grid home architecture] → ACHIEVE [smart grid home architecture]
- ENABLE [secure meter data] → ACHIEVE [secure meter data]

An important part of the goal modeling is when the taxonomy is used to refer to how goals might be linked to each other. The taxonomy is helpful in finding different aspects of the relationships among goals. Some goal links in the taxonomy were found to show goals related to the problem they are intended to avoid or alleviate. For example, in the part of the taxonomy shown in Fig. 6, the goals INCREASES [more cost] and AIM [low cost] are related. *Increasing cost* of electricity is a problem for users that they would want the smart metering system to alleviate. Therefore *lowering cost* is one of the goals of the system. These relationships and the context had to be identified manually. Without manual intervention, the goals would have been interpreted as contradictory. As such, the domain understanding and clarification of the context of the respective goals have resolved the contradictory meaning.

Other relationships represent potential contribution of one goal to another. Referring back to Fig. 6 and continuing with the goal AIM [low cost], the goal INCREASE[off-peak demand] is related to the original problem of *increasing cost*. This relationship shows how increasing off-peak demand, in other words shifting energy use towards off-peak times, helps lower electricity cost (combat the increasing electricity cost). Again, this inference had to be derived manually based on the domain context and the understanding of the semantic relationship between the goals.

Fig. 7 demonstrates these and other goals using goal model diagrams. The notations of the diagrams show contribution links (arrows) between different behavioral goals (parallelograms) and soft goals (clouds). In addition, the dashed ovals surrounding a group of goals represents goals that are closely related by the taxonomy.

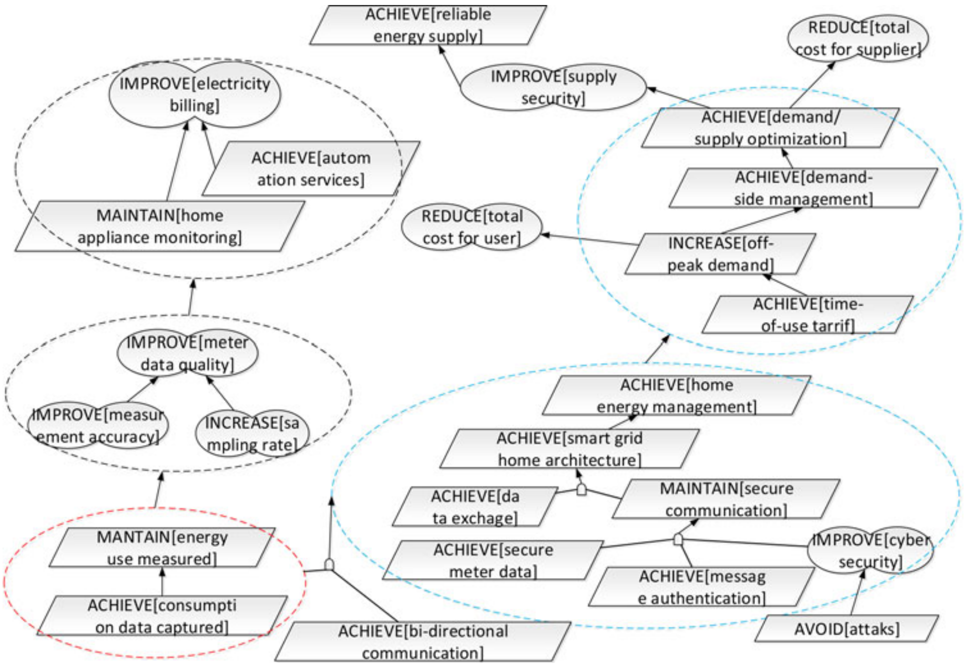


Fig. 7. Functional goal model.

Another example taken from the taxonomy, the goal ACHIEVE [smart grid home architecture] is found linked with goals related to data communication. The parts of Fig. 7 marked with blue dashed ovals shows these goal contributions.

The taxonomy also provides information about different aspects of the system quality goals which should not be overlooked while focusing on solving the problems (such as privacy, or more generally non-functional goals). Non-functional goals such as *privacy* and *stability* are presented by the taxonomy that shows their different aspects. As shown in Fig. 8, highlighted with red dashed ovals *electric grid privacy*, *information privacy* and *user privacy* appear on the taxonomy link related to each other, as do the goals *electric grid stability*, *electric system stability* and *network stability*. In addition, Fig. 8 shows how all goals, including the ones shown in Fig. 7, contribute to the highest-level goal.

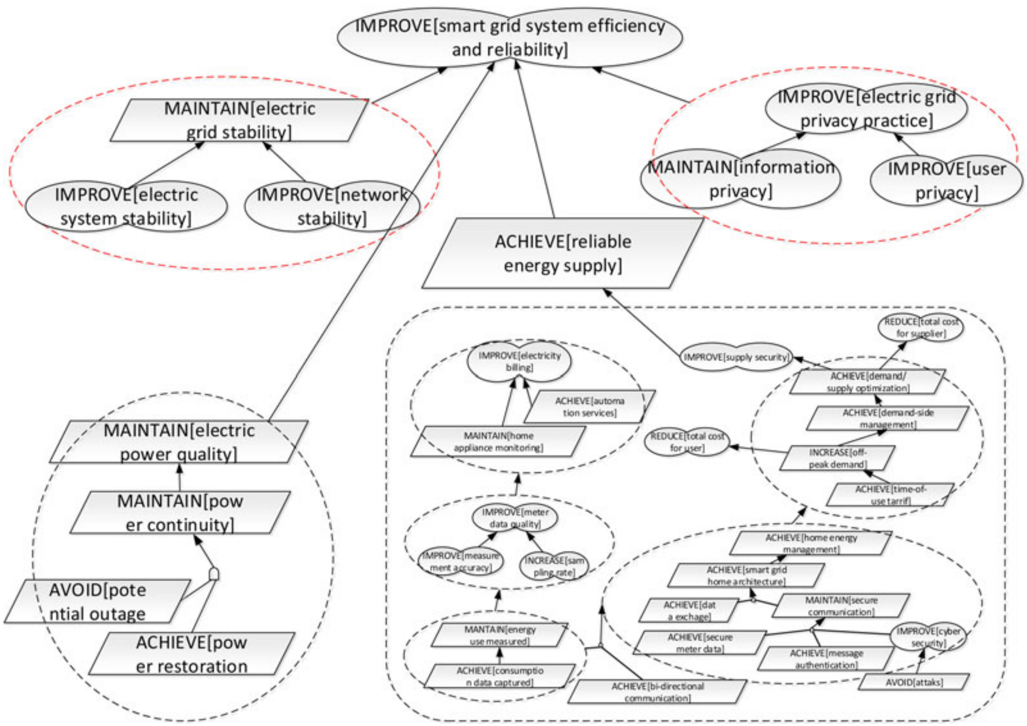


Fig. 8. Non-functional goal model.

Overall, this whole process of making the goal models confirm to the KAOS constructs had to be done manually. Future research on the elicitation of the semantics of the goal models from the text, and their conflicts, might automate up to a certain point this whole step.

## 7 Contributions, Limitations, and Validity Threats

The theoretical contribution of this paper is the extension of bibliometrics approach to goal elicitation from research publications. Research publications, such as conference proceedings, journal papers, and patents, record the state of the art of a technology field of interests through their published research findings and results. However, acquiring and analyzing such knowledge is difficult and time consuming for the increasing amount of text data available in these sources. Computational approaches of language engineering help automate and make more efficient the information retrieval process. Bibliometrics has been used successfully in a variety of applications which include innovation forecasting [50], patent mining [51], literature-based discovery [52], technology forecasting [43], [53], [54], and competitive intelligence [55]. Merging bibliometrics with goal analysis, as we have done in this paper, provides a new perspective not just towards the development of software systems, but also provides a way of approaching innovation forecasting, patent mining, competitive intelligence analysis in terms of goals, all of which are required for strategic systems development.

Another theoretical contribution is that we have demonstrated the feasibility of obtaining useful goal models using just abstract analysis as compared to the full text analysis. Abstracts are short and precise descriptions of the contents of the respective publications. They require the authors to summarize in a synthetic manner the contents of the publication, and the main concepts of the topic of interest. Important high-level abstractions and their inter-relationships are likely to be emphasized. Abstracts have shown to be a sufficient source of information to capture the main points of a domain of interest [35], [43]. It was not clear if the abstract were sufficient source of goal information, which this paper showed otherwise. Moreover, we confirmed the previous finding that the important goals (as it was the case with concepts) are most likely to appear within a single sentence due to the restricted length of the abstracts. This helped us use the NLP algorithms with simple lexico-syntactic rules to extract goals from single sentences in our case study. In the full text, an author has the freedom to discuss each goal across more sentences. Thus, each goal is harder to identify by the extraction algorithm. Finally, besides the computational advantage given by their length, abstracts are resources which are freely available in a format easy to capture and process (plain text or HTML).

Through combination of bibliometrics and extensive up-front goal analysis, NLP-KAOS provides us with the necessary tools to further tackle general theoretical problems that necessitate the shift of focus in the RE strategies from *the solution that the system is expected to provide* towards *what the actual problem is* [56], [57]. The RE strategies need to shift the focus from *systems* to *potential challenges* characterized by being problem focused, predictive, adaptive and holistic [57]. Relatively low-cost NLP-KAOS, as compared to purely manual analysis, can help us maintain a holistic goal model vision of not just the system, but also of its environment, and better control the future evolution of the system and the strategic direction and competitiveness.

As for the applicability of the results, different system design and development activities can benefit from the results of NLP-KAOS. NLP-KAOS produces goals at the high-level (business/organization) as well as at the low-level (technical). It can be used at the early stages of RE to identify potential high-level goals that might provide, e.g., strategic market positioning advantages for the future releases of the respective products. These goals can be also used, e.g., at the acceptance testing stages to check for any missed goals by the developed system. NLP-KAOS captures also low-level technical goals. These goals might include, e.g., advancements in the particular algorithms development or identification of new related technology advancements.

Goals are captured in the form consistent with KAOS (*goal type[goal name]*). NLP-KAOS also relates goals using a taxonomy to make related goals identification easier. However, some goals at the lower-level may seem vague or too general without the detailed understanding of the domain. This is especially evident in the case of complex systems because of the presence of the different systems that cooperate with each other in order to achieve the goals.

Out of all the goals that were specified, one could make an assumption that the goals not appearing in the intersection (of automatically and manually elicited goals) are less significant. This is not the case. The only characterization of the goals that we can make based on this study is that the goals that appear in the intersection are the ones that are relatively validated through their existence in the implemented early models of smart meters. Their validity will be ultimately judged by the market forces and their ultimate propagation in the future generations of smart meters. Likewise, the goals that were detected but are not appearing in the intersection might end up being more relevant in the future generations of smart meters. Thus, we cannot make an ultimate judgment on the validity of any of the goals, but the relative size of the intersection of the goal sets does confirm that we can capture valid goals through the analysis of the research publications.

There are a number of validity threats in this case study. The most common sources of validity threats in a case study are: researchers' background, perception, educational background, surrounding environment, researcher's personal interpretation of the problem under investigation, collected data, findings, and conclusions. Common weakness of case study research are that case studies tend to be subjective, difficult to generalize, validate and repeat. Often, in a case study research, there is significant amount of unnecessary information that contributes to bias and repeatability problems. In order to deal with these problems, we approached and reported the results and findings in a systematic, step-by-step fashion. In addition, to deal with the important issue of the impact and the quality of the work performed by the human analyst, we used a combination of checks:

- following independently developed KAOS method, as opposed to developing and using our own goal elicitation method;
- performing the study within a context of a set of general qualitative evaluation criteria in order to reduce subjectivity;
- ensuring that each project task is performed by a team member with the least likelihood of bias; and
- having each produced artifact verified by an independent team expert.

Given that this is a single case study, it is not possible to judge at this point how critical the impact of the human analyst is (beyond the insurance provided by having different analysts that we used and the independent domain expert for the verification of the results). As such, it is critical to conduct future studies that move beyond the demonstration of the feasibility as in this paper, to various improvements in the used techniques, and through collective research effort reduce and measure the amount of impact of the human analysts.

KAOS was chosen as a relatively mature and well accepted goal specification method with a clear process description [3]. Each step of the manual analysis of the goals was performed within the context of the qualitative evaluation criteria [37]: *naturalistic inquiry*, *emergent design flexibility*, *sampling*, *focus on priorities*, and *holistic perspective*. The RE masters degree student who has performed the manual analysis did not have KAOS or smart grid domain experience. The results were evaluated by a team member who had both KAOS and smart grid experience.

We have used only informal specification of the goals. In this study, we were not able to derive sufficient information needed to perform temporal analysis from the manual analysis of the research publications. However, we have shown that obtaining informal models is feasible (a research contribution). The usefulness of the informal goal models suggests that we could obtain similar results using other GORE approaches that do not rely upon the KAOS particularities and formalizations. Some of the goals defined informally in natural language could be formalized using temporal logic as explained in [3]. The formal definition of goals provides information about objects that exist in a system. Future studies should resolve this weakness of not being able to formalize goals in this case study. Moreover, it will be necessary to demonstrate that the operationalizations are clearly understood and sufficiently refined even without the use of formalizations. We believe that the formalization is ultimately a part of the language itself, rather than the domain, i.e., we should be able to understand and share knowledge about goals even if we are not using a formal modeling language.

The NLP-KAOS study was performed by another team member with data mining and RE experience, but no KAOS experience. The data mining algorithms and results were evaluated by the fourth team member, and all results and comparisons were evaluated by the fifth team member with technical power and smart grid expertise, but no RE expertise. Nevertheless, since this is the first evaluation of a method applied to RICS, it is important to completely mitigate validity threats due to the repeatability of the study on different systems, repeatability with other GORE approaches, and the application on the larger systems in order to deal with the issues of the scalability of the proposed approach. However, this must be done by completely independent teams.

Another validity threat is that we were able to identify only the positive contributions in the intersection of the goal models, without any negative contributions among softgoals. This was pointed out to be somewhat counterintuitive and worrisome, as it appears to suggest the particular smart metering system goal model obtained from the research publications has no weaknesses. We believe this was due to the fact that we were able to identify primarily high-level goals as part of the preliminary manual analysis.

Finally, the resistance of KAOS to engineers' bias depends on the amount and extent of available information resources. At the high-level goal identification and refinement it is quite resistant if the stakeholders are clear about their goals. For lower-level goals and responsibility assignments it may give more freedom to an analyst to make design decisions and may be less objective. However, KAOS provides support for requirements validation and traceability. This could limit the freedom and bias of the analyst in the decision making. This particular research work was done entirely based on the available research publications, and without any prior knowledge of the smart metering system. This helped further reduce bias, and the credibility of the results is supported by the sources of the information.

## 8 Conclusions and Future Work

This paper has presented a novel approach, NLP-KAOS, based on data mining and NLP techniques, which aims to make more efficient the task of early stages of goals elicitation. A semi-automatic mining process of text analysis helps the requirements engineer to gather information from abstracts of research publications by extracting goals from text and organizing them in a taxonomy. NLP-KAOS was applied and evaluated on a smart metering system case study. The smart metering system is an important subsystem of the smart grid, a novel concept of electrical infrastructure which is an example of a RICS. The analysis showed the advantages of using NLP-KAOS, and compared it to the results obtained using the manual KAOS approach. Furthermore, the case study showed that it is feasible to extract a useful set of goals from research publications.

We believe that a potential future, theoretical research direction could be linguistic research into how one can write research papers in a way that makes it easier to perform automatic data mining of not just goals, but requirements, concepts, constraints, etc. Another direction is the development of formal reasoning process for deriving requirements from goals captured from research publications in order to perform strategic analysis of future business and technological directions. This would make RE work a subset of general strategic development that most of data mining is currently used for.

## ACKNOWLEDGMENTS

Davor Svetinovic is the Corresponding Author.

## REFERENCES

[1] A. I. Antón and C. Potts, "The use of goals to surface requirements for evolving systems," in *Proc. IEEE Int. Conf. Softw. Eng.*, Apr. 1998, pp. 157–166.

[2] A. Dardenne, A. van Lamsweerde, and S. Fickas, "Goal-directed requirements acquisition," *Sci. Comput. Program.*, vol. 20, no. 12, pp. 3–50, 1993.

[3] A. Van Lamsweerde, "Goal-oriented requirements engineering: A guided tour," in *Proc. 5th IEEE Int. Symp. Requirements Eng.*, 2001, pp. 249–262.

[4] A. Van Lamsweerde, "From system goals to software architecture," in *Formal Methods for Software Architectures*. New York, NY, USA: Springer-Verlag, 2003, pp. 25–43.

[5] D. Klein and C. D. Manning, "Accurate unlexicalized parsing," in *Proc. 41st Annu. Meeting Associat. Comput. Linguistics*, vol. 1, 2003, pp. 423–430.

[6] D. Klein and C. D. Manning, "Fast exact inference with a factored model for natural language parsing," in *Proc. Adv. Neural Inf. Process. Syst. 15*, 2003, pp. 3–10.

[7] E. Maedche and S. Staab, "Ontology learning for the semantic web," *IEEE Intell. Syst.*, vol. 16, no. 2, pp. 72–79, Mar./Apr. 2001.

[8] P. Heymann and H. Garcia-Molina, "Collaborative creation of communal hierarchical taxonomies in social tagging systems," Comput. Sci. Dept., Standford Univ., Stanford, CA, USA, Tech. Rep. ilprints775, Apr. 2006.

[9] A. Van Lamsweerde and L. Willemet, "Inferring declarative requirements specifications from operational scenarios," *IEEE Trans. Softw. Eng.*, vol. 24, no. 12, pp. 1089–1114, Dec. 1998.

[10] A. Van Lamsweerde, R. Darimont, and E. Letier, "Managing conflicts in goal-driven requirements engineering," *IEEE Trans. Softw. Eng.*, vol. 24, no. 11, pp. 908–926, Nov. 1998.

[11] A. Van Lamsweerde and E. Letier, "Handling obstacles in goal-oriented requirements engineering," *IEEE Trans. Softw. Eng.*, vol. 26, no. 10, pp. 978–1005, Oct. 2000.

[12] A. Van Lamsweerde, *Requirements Engineering—From System Goals to UML Models to Software Specifications*. New York, NY, USA: Wiley, 2009.

[13] F. Zickert, "Evaluation of the goal-oriented requirements engineering method kaos," in *Proc. Amer. Conf. Inf. Syst.*, 2010, pp. 1–9.

[14] E. Letier, "Reasoning about agents in goal-oriented requirements engineering," Ph.D. dissertation, Université catholique de Louvain, Louvain-la-Neuve, Belgium, 2001.

[15] L. Kof, "Natural language processing for requirements engineering: Applicability to large requirements documents," in *Proc. 19th Int. Conf. Autom. Softw. Eng. Workshops*, 2004, pp. 91–102.

[16] K. Ryan, "The role of natural language in requirements engineering," in *Proc. IEEE Int. Symp. Requirements Eng.*, 1993, pp. 240–242.

[17] P. Sawyer, P. Rayson, and K. Cosh, "Shallow knowledge as an aid to deep understanding in early phase requirements engineering," *IEEE Trans. Softw. Eng.*, vol. 31, no. 11, pp. 969–981, Nov. 2005.

[18] H. Happel and S. Seedorf, "Applications of ontologies in software engineering," in *Proc. Workshop Sematic Web Enabled Softw. Eng.*, 2006, pp. 5–9.

[19] K. Breitman and J. do Prado Leite, "Ontology as a requirements engineering product," in *Proc. 11th IEEE Int. Requirements Eng. Conf.*, Sep. 2003, pp. 309–319.

[20] T. R. Gruber, "A translation approach to portable ontology specifications," *Knowl. Acquisition*, vol. 2, pp. 199–220, 1993.

[21] Y. Zhao, J. Dong, and T. Peng, "Ontology classification for semantic-web-based software engineering," *IEEE Trans. Serv. Comput.*, vol. 2, no. 4, pp. 303–317, Oct. 2009.

[22] H. Kaiya and M. Saeki, "Using domain ontology as domain knowledge for requirements elicitation," in *14th IEEE Int. Conf. Requirements Eng.*, Sep. 2006, pp. 189–198.

[23] I. Jureta, J. Mylopoulos, and S. Faulkner, "Revisiting the core ontology and problem in requirements engineering," in *Proc. 16th IEEE Int. Requirements Eng.*, Sep. 2008, pp. 71–80.

[24] W. Wong, W. Liu, and M. Bennamoun, "Ontology learning from text: A look back and into the future," *ACM Comput. Surv.*, vol. 44, no. 4, pp. 20:1–20:36, Sep. 2012.

[25] L. Goldin and D. M. Berry, "Abstfinder, a prototype natural language text abstraction finder for use in requirements elicitation," *Autom. Softw. Eng.*, vol. 4, no. 4, pp. 375–412, 1997.

[26] R. Gacitua, P. Sawyer, and V. Gervasi, "Relevance-based abstraction identification: Technique and evaluation," *Requirements Eng.*, vol. 16, no. 3, pp. 251–265, 2011.

[27] P. Buitelaar and B. Magnini, "Ontology learning from text: An overview," in *Ontology Learning from Text: Methods, Applications and Evaluation*, P. Paul Buitelaar, P. Cimiano, and B. Magnini, Eds. Amsterdam, The Netherlands: IOS Press, 2005, pp. 3–12.

[28] F. Giunchiglia and I. Zaihrayeu, "Lightweight ontologies," *Encyclopedia Database Syst.*, pp. 1613–1619, 2007, 10.1007/978-0-387-39940-9_1314.

[29] F. Fürst and F. Trichet, "Heavyweight ontology engineering," in *Proc. Move Meaningful Internet Syst.: OTM 2006 Workshops*, 2006, pp. 38–39.

[30] M. Hearst, "Automatic acquisition of hyponyms from large text corpora," in *Proc. 14th Conf. Comput. Linguistics*, 1992, pp. 539–545.

[31] G. Miller, "Wordnet: A lexical database for English," *Commun. ACM*, vol. 38, no. 11, pp. 39–41, 1995.

[32] R. Navigli, P. Velardi, and A. Gangemi, "Ontology learning and its application to automated terminology translation," *IEEE Intell. Syst.*, vol. 18, no. 1, pp. 22–31, Jan./Feb. 2003.

[33] M. Sanderson and B. Croft, "Deriving concept hierarchies from text," in *Proc. 22nd Annu. Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 1999, pp. 206–213.

[34] A. Porter, "How 'tech mining' can enhance R&D management," *Res. Technol. Manage.*, vol. 50, no. 2, pp. 15–20, 2007.

[35] A. Porter and S. Cunningham, *Tech Mining: Exploiting New Technologies for Competitive Advantage*. New York, NY, USA: Wiley-Interscience, 2005.

[36] S. Cunningham, A. Porter, and N. Newman, "Special issue on tech mining," *Technol. Forecasting Social Change*, vol. 73, no. 8, pp. 915–922, Oct. 2006.

[37] S. Woldeamlak. KAOS for research-intensive complex systems: Smart metering case study evaluation. Master's thesis, Masdar Inst. Sci. Technol., Masdar City, United Arab Emirates. (2013) [Online]. Available: http://goo.gl/xRzm0t

[38] J. W. Creswell, *Research Design: Qualitative, Quantitative, and Mixed Methods Approaches*. Newbury Park, CA, USA: Sage, 2013.

[39] V. Vaishnavi and W. Kuechler. Design research in information systems. (2004) [Online]. Available: http://goo.gl/ReiyVM

[40] A. Van Lamsweerde, *Requirements Engineering: From System Goals to UML Models to Software Specifications*. New York, NY, USA: Wiley, 2009.

[41] D. Rusu, L. Dali, B. Fortuna, M. Grobelnik, and D. Mladenic, "Triplet extraction from sentences," in *Proc. 10th Int. Multiconf. Inf. Soc.*, 2007, pp. 8–12.

[42] D. Rusu, B. Fortuna, M. Grobelnik, and D. Mladenic, "Semantic graphs derived from triplets with application in document summarization," *Informatica J.*, vol. 33, no. 3, pp. 357–362, 2009.

[43] A. Henschel, E. Casagrande, W. Woon, I. Janajreh, and S. Madnick, "A unified approach for taxonomy- based technology forecasting," in *Business Intelligence Applications and the Web: Models, Systems and Technologies*. Hershey, PA, USA: IGI Global, 2010, pp. 178–197.

[44] A. Henschel, W. Woon, T. Wachter, and S. Madnick, "Comparison of generality based algorithm variants for automatic taxonomy generation," in *Proc. Int. Conf. Innovations in Inf. Technol.*, 2009, pp. 160–164.

[45] Z. Harris, *Mathematical Structures of Language*. New York, NY, USA: Wiley, 1968.

[46] H. Suleiman, K. Ahmed, N. Zafar, E. Phillips, D. Svetinovic, and O. de Weck, "Inter-domain analysis of smart grid domain dependencies using domain-link matrices," *IEEE Trans. Smart Grid*, vol. 3, no. 2, pp. 692–709, Jun. 2012.

[47] H. Suleiman and D. Svetinovic, "Evaluating the effectiveness of the security quality requirements engineering (square) method: A case study using smart grid advanced metering infrastructure," *Requirements Eng.*, vol. 18, no. 3, pp. 251–279, 2013.

[48] G. T. Heydt, "The next generation of power distribution systems," *IEEE Trans. Smart Grid*, vol. 1, no. 3, pp. 225–235, Dec. 2010.

[49] N. Zafar, E. Arnautovic, A. Diabat, and D. Svetinovic, "System security requirements analysis: A smart grid case study," *Syst. Eng.*, vol. 17, no. 1, pp. 77–88, 2014.

[50] R. J. Watts, A. L. Porter, and N. C. Newman, "Innovation forecasting using bibliometrics," *Competitive Intell. Rev.*, vol. 9, no. 4, pp. 11–19, 1998.

[51] Y.-H. Tseng, C.-J. Lin, and Y.-I. Lin, "Text mining techniques for patent analysis," *Inf. Process. Manag.*, vol. 43, no. 5, pp. 1216–1247, Sep. 2007.

[52] R. N. Kostoff, "Systematic acceleration of radical discovery and innovation in science and technology," *Technol. Forecasting Social Change*, vol. 73, no. 8, pp. 923–936, Oct. 2006.

[53] T. U. Daim, G. Rueda, H. Martin, and P. Gerdsri, "Forecasting emerging technologies: Use of bibliometrics and patent analysis," *Technol. Forecasting Social Change*, vol. 73, no. 8, pp. 981–1012, 2006.

[54] J. Martino, "A review of selected recent advances in technological forecasting," *Technol. Forecasting Social Change*, vol. 70, no. 8, pp. 719–733, 2003.

[55] M.-J. Shih, D.-R. Liu, and M.-L. Hsu, "Discovering competitive intelligence by mining changes in patent trends," *Expert Syst. Appl.*, vol. 37, no. 4, pp. 2882–2890, 2010.

[56] G. Polacek and D. Verma, "Requirements engineering for complex adaptive systems: Principles vs. rules," in *Proc. 7th Annu. Conf. Syst. Eng. Res.*, Apr.2009, pp. 1–7.

[57] D. Svetinovic, "Strategic requirements engineering for complex sustainable systems," *Syst. Eng.*, vol. 16, no. 2, pp. 165–174, 2013.

**Erik Casagrande** received the MSc degree in control engineering in 2003 and the MSc degree in numerical modeling in 2005 from Padova University, Italy. He received the PhD degree from the Neural Computing Research Group, Aston University, Birmingham, United Kingdom, in 2010. He is currently a research scientist with the Masdar Institute of Science and Technology, Abu Dhabi, United Arab Emirates. His research interests include machine learning, data mining and time series techniques for biomedical data, power systems and smart grids, natural language processing, and atmospheric science. He is a member of the IEEE.

**Selamawit Woldeamlak** received the BSc degree in computer science from the University of Asmara, Asmara, Eritrea, in 2004, and the MSc degree in computing and information sciences from Masdar Institute of Science and Technology, Abu Dhabi, United Arab Emirates, in 2013. Her research interests include strategic requirements engineering with an emphasis on complex (sustainable) systems.

**Wei Lee Woon (M'08)** received the BEng degree in electronic engineering (Hons.) from the University of Manchester Institute of Science and Technology, Manchester, United Kingdom (now merged with the University of Manchester), in 1997, and the PhD degree from the Neural Computing Research Group, Aston University, Birmingham, United Kingdom, in 2002. Upon graduation, he joined the Malaysia University of Science and Technology (MUST) as an assistant professor, where he served until 2007. He subsequently joined the Masdar Institute of Science and Technology, Abu Dhabi, United Arab Emirates. He was also a visiting researcher at the Massachusetts Institute of Technology, Cambridge, and at the RIKEN Brain Science Institute, Tokyo, Japan. His research interests include technology mining, analysis of distributed generation systems, and EEG signal analysis. He is a member of the IEEE.

**H.H. Zeineldin (M'06-SM'13)** received the BSc and MSc degrees in electrical engineering from Cairo University, Cairo, Egypt, in 1999 and 2002, respectively, and the PhD degree in electrical and computer engineering from University of Waterloo, Waterloo, ON, Canada, in 2006. He was with Smith and Andersen Electrical Engineering, Inc., where he was involved with projects involving distribution system design, protection, and distributed generation. He then joined the Massachusetts Institute of Technology, Cambridge, as a visiting professor. He was with the Faculty of Engineering, Cairo University and currently, he is an associate professor with the Masdar Institute of Science and Technology, Masdar City, UAE. He is currently an editor of the *IEEE Transactions on Energy Conversion* and the *IEEE Transactions on Smart Grids*. His current interests include power system protection, distributed generation, and microgrids. He is a senior member of the IEEE.

**Davor Svetinovic** received the MMath and PhD degrees in 2002 and 2006, respectively, in computer science from the University of Waterloo, Canada. Previously, he was a visiting researcher at Massachusetts Institute of Technology (MIT), US, and he worked as a postdoctoral researcher at Lero – the Irish Software Engineering Center, Ireland, and Vienna University of Technology, Austria. His current research interests include: software engineering, strategic requirements engineering, systems architecture with emphasis on smart grids, and sustainable development from the systems security perspective. He is a member of the IEEE.

▷ **For more information on this or any other computing topic, please visit our Digital Library at** www.computer.org/publications/dlib.