Literature Reviews

Goal Modeling Techniques in Requirements Engineering: A Systematic Literature Review

Iman Ahmed ElSayed, Zeinab Ezz and Eman Nasr

Department of Computer Science, Institute of Statistical Studies and Research, Cairo University, Cairo, Egypt

Abstract: Goal modeling techniques plays a crucial role in requirements engineering since they facilitate the reach of requirements to stakeholders in an understandable easy manner and also in a professional well defined pattern to developers. Modeling of goals are encouraged and proposed during requirements elicitation in order to detail, understand and describe problems associated with current organizational structures and behavior. However, a current challenge appears which is how to manage modeling of goals if either these goals were elicited in early or late phases? Eventually if that occurred the rest of the challenge comes in how to model these goals in the presence of uncertainty? This paper presents a systematic literature review of the current goal modeling techniques dealing with both early and late requirements such as: i* framework, tropos, GRL and UML. The results for research in this study show that although there are models for both early and late requirements, most techniques to a great extent are used for modeling early requirements. The findings lead us to identify two future work elicited in the study that might help a lot in modeling goals.

Keywords: Goal Modeling Techniques, Early and Late Requirements, Goal Oriented Approaches, Context in Goal Modeling Techniques, Systematic Literature Review

Introduction

Requirements Engineering (RE) has been established as a separate field for investigation, definition, analysis and experimentation and in return the definition of RE has appeared (Kavakli and Loucopoulos, 2003). Initially RE has been defined as requirements needed to assess and measure the current concerned software implementation. Later on through years the boundaries for the RE definition has been extended to include not only the software implementation but also the whole organizational structure including its systems (Kavakli and Loucopoulos, 2003). Eventually we can say that RE is the part of software engineering which looks into handling problems occurring and taking place in organizational settings since it is concerned mainly with problems occurring from business goals, plans and systems-to-be in the organizational objectives. A question at this point appeared which was "How RE will manage multi-stakeholders' criteria while considering also coordinating their actions in order to achieve required results?". The following question was the reason for the rise of the principle of goal oriented approach.

A variety of criteria is used in order to decide the degree of success of a certain system but the main criteria for assessing this system is to what extent this system meets the purpose it was done for in first place. As a result for this principle that has been assessed, goal oriented approaches have increased a lot lately. However, this increase in goal oriented approaches was due to many reasons such as the inadequacy of traditional systems with more and more complex systems, most techniques focus on modeling and specification of the software alone, also most of the time non-functional requirements are left outside from the specifications and finally, traditional techniques don't allow alternative system configurations and so in return goal oriented approaches and goal modeling techniques have been given a lot of weight in the requirements engineering field lately.

This paper presents a comprehensive review of scientific papers that reflects the usage of different goal modeling techniques in eliciting both early and late requirements (goals). In addition to the above, handling uncertainty in goal modeling is considered an extra challenge too. An example from a current

research and practice; that dealt with the uncertainty point in goal modeling and how to handle it which lead later to some limitations that opened up new future models which might help in dealing with the limitations; is used in this study.

The objective of this work is achieved through a Systematic Literature Review (SLR) in order to provide an objective and balanced summary of research results found. A SLR research is a technique in which the state of the art of a particular field of knowledge is analyzed and discussed through defining a problem statement which is concerned with the research objective, research questions' sources of information, search string, the criteria for inclusion and exclusion for sources found from the search performed. Various SLR have been done in the field of RE such as (Kavakli and Loucopoulos, 2003).

The paper is structured as follow: Section 2 discusses the main design of a SLR. Section 3 discusses the results. Section 4 discusses a paper (Torkar *et al.*, 2009) as an example for discussing uncertainty in goal modeling. Limitations and threats that can be realized from this SLR and the example paper used are discussed in section 5. Finally, section 6 presents conclusions and future work.

Planning and Designing the SLR

The protocol that we follow is summarized in this section. Section 2.1 gives a scope on the research objective from this SLR. Section 2.2 presents the Research Questions (RQs) that guide the SLR. Section 2.3 presents our sources selection. In section 2.4 we show the identification and selection of our primary studies. Section 2.5 states the inclusion and exclusion criteria. Section 2.6 presents the data extraction strategy that is used for this research followed finally by section 2.7 which will state the results and conduct the review.

Research Questions

The overall objective of this research is stated in section 1, but in this section it will be precisely defined and much more refined in order to focus on it more. The interest in this SLR is in studying the meaning of goals and their different types, showing the importance of goal modeling in requirements engineering and when exactly to use them, studying what are goal oriented approaches and how they are used in both early and late requirements modeling after knowing the difference between both types of requirements exactly and when to use to use each of them, studying specifically early requirements goal modeling techniques; since most of goal modeling

techniques concentrate on early requirements modeling; and finally study the analysis for each of the chosen early requirement goal modeling techniques.

The RQ that we plan on answering in this SLR are as follow:

- RQ1: What are goals and their types?
- RQ2: What is the goal modeling concept?
- RQ3: Why and when to use goal models?
- RQ4: What are early and late requirements? And what is the difference between them both?
- RQ5: What are the goal modeling techniques for both types of requirements?
- RQ6: What are goal oriented approaches?
- RQ7: What is the most known late requirements goal modeling technique?
- RQ8: What are the current available analysis goal satisfaction procedures? And how analysis takes place in each of the early requirements goal modeling techniques?
- RQ9: What are the most known early requirements goal modeling techniques? And what are the tools supporting each?
- RQ10: Can context be handled in these techniques? If yes, how?

Source Selection

Two types of search methods have been used to select and choose appropriate papers related to our RQ in order to perform this SLR. The first type of search was the automatic search and these were mainly from:

- IEEE international conferences
- ACM digital library
- John Wiley and Sons publications (Books)
- Elsevier

The well known journals, workshops and conferences concerned with requirements engineering were searched such as: Requirements Engineering Journal, Software Engineering and Knowledge Engineering international conference, Advances in Conceptual Modeling Conference, Proceedings of the 8th CAiSE/IFIP8, III Workshop de Engenharia de Requisitos 1ed. Brasil: Rio de Janeiro, Anais WER and University of Toronto Quarterly Journal.

In addition to all of the above, another type of search has been used; a chosen paper which handles one of the challenges that faces early requirements decision making which is presence of uncertainty has been discussed with the authors through emails.

Table 1. Number of selected items from each source

| Source | Automatic search | | | | | | | | Manual Search | |
	IEEE international Conferences	ACM digital Library	John Wiley and Sons	El sevier	Springer	Other international Journals	Other international conferences	International	Emailing author	Total Results
Selected results	6	4	1	1	3	3	2	1	3	24

Source Identification Strings of Primary Studies

Based on the research goals, all selected searches are based on defining search string to identify our study. The search string is defined into four partitions: The first partition states the different definitions and types of goals, the main principles of goal modeling and when and why goal modeling is to be used, the second partition links the first partition with the definition of both early and late requirements, the difference between both of them and different goal oriented approaches related to both early and late requirements, in the third partition the most known approaches for early requirements are elicited with their analysis, finally a question in the last partition is asked which is can context in early requirements engineering techniques be handled or not and a suggestion for a requirement recommender system is given. The search strings defined is as follow:

- Goals OR goal modeling OR History of Goal Modeling in requirements engineering OR goal modeling technique OR goal modeling for soft goals OR goal oriented approaches OR goal oriented approaches versus goal modeling OR History of goal oriented approaches) AND (Early phase requirements OR Late phase requirements OR late requirements analysis techniques OR difference between early and late requirements techniques)
- Early requirement goal modeling technique) AND (Uncertainty in goal modeling OR uncertainty in early requirements modeling) OR(Analyzing early requirement goal modeling technique OR goal satisfaction analysis procedure
- Context in early requirement goal modeling technique) AND (Requirement engineering recommender system

Inclusion and Exclusion Criteria

Papers that have been included and taken into consideration were those papers that their title seems to have relevance our SLR and related to our search. Papers published in conference proceedings, workshops and scientific journals between 1998 and 2014 were considered to be highly significant to the SLR. As for the exclusion criteria, we excluded any search output irrelevant to our search string and disagree with the inclusion criteria in addition to any papers not written in English.

Conducting the Review

In this stage we're starting to state from which sources our primary study has been conducted and established, Table 1 represents the final set of relevant papers chosen from each source as follow.

Results

After finishing the initial stage of the selection process, 21 papers were included plus 3 emails with authors in order to vision some missing points required.

The results for RQ1 which is defining a goal resulted in finding many definitions for goals in the RE field but they all ended into two major definitions for the word goal according to the point of view of two major researchers which are Van Lamsweerde which stated that goals are objectives achieved by the system through cooperation of agents in the software-to-be and in the environment (Horkoff *et al.*, 2014) and the other definition was mentioned by Anton stating that a goal is a high level objective of the business, organization or system that helps in eliciting and realizing the reasons why a system is needed and according to this, decisions are taken within the organization (Horkoff *et al.*, 2014; VAn Lamsweerde, 2001). Although these are considered two separate definitions, but it is realized that the first definition of that of Lamsweerde includes the second one defined by Anton, since achieving high level objectives are objictifies achieved by involving many agents from the organization in the software-to-be which will automatically lead to decision making in the enterprise.

The results for trying to realize the difference between a goal and a requirement which is criteria 2 led to understanding that both words are not totally the same and that a requirement is considered to be a subset word from the word goal; since, a goal is an objective achieved by the system through cooperation of agents in the software to be while a requirement is a goal under the responsibility of a single agent (Horkoff *et al.*, 2014). In conclusion, requirements are small goals achieved by single agents in order to achieve the major goal implemented for it the whole software-to-be with the participation of other agents for other requirements.

As for results for criteria 3 which is different types of goals, it was found that many researchers have categorized goals. Goals have been categorized to be either soft or hard goals, which is considered to be the well known

categorization for goals. Also goals have been categorized as functional and non-functional goals; since these are the major known types of requirements and requirements are considered to be a single agent goal; so in return goals have been categorized according to them too. The last categorization for goals was defining goals into soft and behavioral goals. Actually, the most categorization for types of goals that has been realized and used since it is more general is the categorization of soft and hard goals. Soft goals are goals which describe something that a stakeholder wants to achieve from the system qualities, these are judged according to stakeholder's criteria of satisfaction and are difficult to quantify. As for hard goals, they describe functions the system will perform, they are either satisfied or not satisfied in the system and as a result they can be judged easily and are easily quantified.

The results for the appearance of goal modeling concept which is criteria 4, it was found that it first appeared when modeling of goals has been proposed during requirements elicitation in order to describe the current organizational behavior, afterwards it has been updated to include also setting the objectives for any changes within the organization in order to fit these changes well. Equally, goal analysis techniques have been used in the context of requirements negotiation in order to assist reasoning about the need for organizational change, the goal model concept has also been used in requirements specification to describe how organizational change can be implemented in terms of the new system's components by relating business goals to Functional Requirements (FR) and Non-Functional Requirements (NFR) of the system, finally it has been used in requirements validation phase to define the stake holders 'criteria against which the fitness of system components is assessed.

As for results conducted from criteria 5 which is why do we use goal modeling concept in first place were elicited stating that goal modeling gives some rationality for requirements (raw information) to be understood well, in addition to this, it helps in identifying stable information, moreover guides requirement elaboration, facilitates understanding and describing problems associated with business structures and processes and their supporting systems. Finally, goal modeling allows and assists in taking a decision between different alternatives which also leads to handling uncertainties (design uncertainties) in different contexts.

The result of benefits of goal modeling and when to use it which is criteria 6 recommends using goal modeling rather than traditional methods that were used in Requirements Engineering (RE) field since it helps in taking a wider system engineering perspective compared to the traditional RE methods which later assessed in adding support for early requirements analysis, in addition to that goal modeling provides a precise criterion for sufficient completeness of a requirements

specification, even if these requirements are complex and in return goal modeling provides a natural mechanism for structuring complex requirements' documents since it provides the basis for detection and management of conflicts among requirements. Goal modeling also helps in capturing variability in the problem domain which leads to an excellent way in communicating requirements to customers. We use goal modeling for capturing both early and late requirements.

The result for the difference between early and late requirements which is criteria 7 is in many points such as: From the analyst point of view, in early requirements the analyst is trying to understand an organizational setting but as for late requirements, the analyst is trying to determine the best configuration of the system and its environment and in return formulates a solution. Early requirements mainly concentrate on the analysis and modeling of the environment for the system-to-be, the organizational context and the stakeholders, their objectives and their relationships. As for late requirements, they mainly concentrate on embedding the system into the organization, identifying and adjusting the boundaries of the system with its environment and identifying the system requirements and assumptions about the environment. In addition to the above, early requirements amount to the definition of a search space and a search among alternatives within that space while late requirements amount to refining, disambiguating and completing the description of the chosen alternative. From the well known early requirements goal modeling techniques are i* framework, Tropos and GRL, while one of the most know late requirements goal modeling techniques is the Unified Modeling Language (UML).

The results for criterion 8 which is stating the target from goal oriented approaches and that is focusing on the description and evaluation of alternatives and their relationship to the organizational objectives. Goal oriented approaches have been linked to the four RE activities as shown in the below (Fig. 1) linking to them goal oriented approaches that can handle requirements within each activity (Horkoff and Yu, 2016).

Examples of both late and early requirements' results have been elicited for criterion 9. One of the most popular examples for late requirements goal modeling techniques is the UML. UML was originally based on the object oriented modeling technique whose aim is to provide a standard way to visualize the design of a system. UML has two types of views: Systematic (or structural) view and dynamic (or behavioral) view. In spite of being one of the most known techniques for modeling late requirements, UML suffers many disadvantages such as: It is so time consuming, we can't identify exactly who benefits from UML, the UML diagram might be overcomplicated for the customer to understand and the emphasis most of the times is mainly on the design which annoys the developer in his work a lot.

RE Activity		Goal Analysis Contribution	Goal-Oriented Approach
• requirements elicitation	1.	understanding the current organisational situation,	GOMS, Goal-based Workflow, *i**, EKD
	2.	understanding the need for change	ISAC, F³
• requirements negotiation	3.	providing the deliberation context of the RE process	SIBYL, REMAP, The reasoning loop model
• requirements specification	4.	relating business goals to functional and non-functional system components	KAOS, GBRAM , the NFR framework, the Goal-scenario coupling frame work
• requirements validation	5.	validating system specifications against stakeholders' goals	GSN, GQM

Fig. 1. Goal oriented approaches' linked with RE activities (VAn Lamsweerde, 2001)

In order to model early requirements properly, agent oriented approach has been defined to analyze such requirements and select the best alternatives that results later in achieving their goals. From the popular early requirements goal models are:

i* Framework

i* framework has been proposed by Horkoff and Yu (2016) which later was discovered for many drawbacks that lead to the appearance of variants of i* which are Tropos and GRL (Yu, 1997).

The i* framework has two ways of representation: The graphical and the formal representations. As for the graphical representation, it has two main modeling components; the Strategic Dependency (SD) model and the Strategic Rationale (SR) model The SD model is concerned with describing mainly the dependency relationship between actors within the organization and as a result, it helps in understanding how a certain goal is embedded within the organization. Therefore, it always answers the "Whys" questions. On the other hand, the SR model is concerned with describing stakeholders' interests and how they might be addressed through various system configurations and stakeholders' evaluation of various alternatives respecting their interests (Ayala *et al.*, 2005; Horkoff and Yu, 2016). I* actors can be categorized into agents, positions and roles. An agent occupies a position, a position covers a role. Consequently an agent plays role covered by a certain position. Actors and their categorization can also be decomposed into another actor through the is-part-of relationship type (Quartel *et al.*, 2009). Actors have some intentional properties such as goal, belief, attributes and commitment (Ayala *et al.*, 2005). I* framework provides a number of analyzing levels in terms of ability, workability, viability and believability (Horkoff and Yu, 2016). i* models and analyzes security and privacy requirement using Secure i* (SI*) (Amyot *et al.*, 2009; Kiyavitskaya and Zannone, 2008).

The i* basic elements (Fig. 1) are:

- The intentional elements: Which are goals, soft goals, tasks, resources of an actor boundary (Yu, 1997; Ayala *et al.*, 2005)
- Links: Which connect the i* model elements together through means-end, task-decomposition or contribution links (positive (Make/Help links), negative (Hurt/Break links) and the unknown (Yu, 1997; Ayala *et al.*, 2005)
- The satisfaction levels (Yu, 1997; Ayala *et al.*, 2005)

Reasoning elements: They are represented through routines, rules and beliefs (Yu, 1997).

For large and complex organizations, the i* graphical representation is not suitable as a result, i* formal notation is used. The formal notation uses the First Order Logic (FOL). Six predicates are used indicating the possible six analysis labels v = {FS (), PS (), FD (), PD (), C (), U ()} where:

FS(): Represents the fully satisfied, PS(): Stands for the partially satisfied, FD(): Is the fully denied label, PD(): Represents the partially denied, C(): Stands for the conflict label and the U(): Is the unknown label. The predicate holds when the label applies. Examples for propagation rules for labels are found in Fig. 2.

i* tools. In spite of the existence of many tools supporting i* language such as OME and REDEPEND, it always suffered from having some strange situations which mainly appears due to the incompleteness of the formalization of i*. The reason for this incompleteness is the existence of many confused situations, e.g., the "is-a" relation is used profusely, however, it is not defined as a i* constructor. Moreover, many incomplete definitions exist, e.g., no indication about the type and the number of roots in the internal decomposition of an actor. In addition to the presences of some ambiguous definitions, e.g., the dependency link must have two different importance degrees each implies one of the involved actor (depender, dependee) (Yu, 1997).
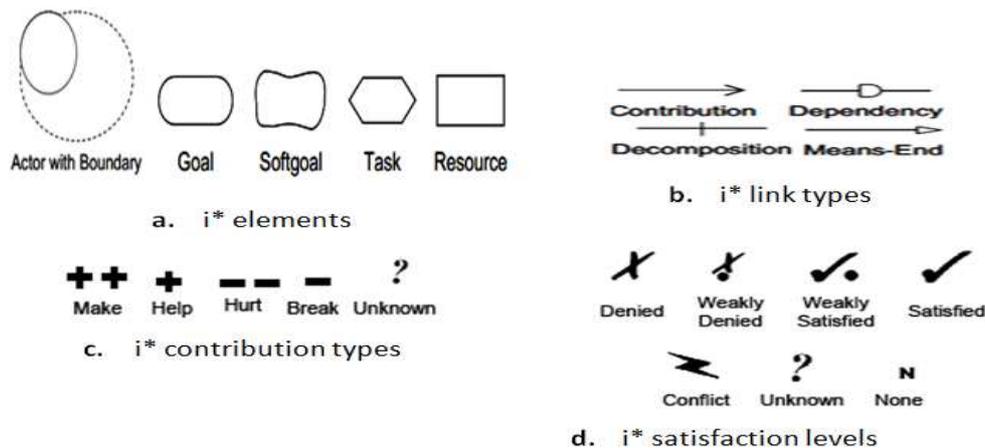
Fig. 2. The i* basic elements

Tropos

Tropos is an agent-oriented software development methodology. It adopts i* model. Tropos supports four phases from the software development which are, early requirements, late requirements, architectural design and detailed design. The early requirement phase is intended to understand the organizational context that the system-to-be will be built on. The Late requirement phase is concerned with defining the system-to-be's functional and non-functional requirements. The architectural design is concerned with defining the system global architecture, whereas the detailed design concerned with defining the behavior of each software component in more details. Tropos can represent organizational goals either graphically or formally (Kiyavitskaya and Zannone, 2008). Tropos models and analyzes security and privacy requirement using secure Tropos (Franch *et al.*, 2016).

Tropos graphical representation has two diagrams for modeling and analysis of organizational requirements and goals which are the actor diagram and the rationale diagram. The actor diagram, similar to SD model in i* while the rationale diagram is similar to SR model in i*.

Tropos basic elements are the same as those of i* differing only in the links types, where Tropos provides AND/OR decomposition link instead of the task-decomposition of i*.

Tropos Formal Representation is preferred to model large and complex organizations requirements. It uses the FOL of that of i*. Tropos forward propagation rules are found in Fig. 3. Backward propagation rules are found in Fig. 4.

Tropos Tools

Tropos forward and backward propagation is supported by the Goal Reasoning Tool(Gr-Tool[1]). The Gr-Tool is a graphical tool for representing goal models and applying the required analysis algorithm (Kiyavitskaya and Zannone, 2008). GOALSOLVE and GOALMINSOLVE tools are implemented to support the backward propagation first and second approach.

Goal-Oriented Requirement Language (GRL)

GRL is an agent-oriented and goal-oriented modeling language that supports reasoning about non-functional requirements and quality attributes. It is influenced by both the i* and the NFR frameworks for specifying non-functional requirements (Yu, 1997). Recently, GRL is standardized by the International Telecommunication Union (ITU-T) as a part of the User Requirements Notation (Quartel *et al.*, 2009).

User Requirements Notation (URN) is used to support all the RE phases. It allows requirement engineers to elicit and specify requirements then analyze such requirements to ensure its completeness and correctness. URN consists of two complementary languages which are Goal-oriented Requirement Language (GRL) and Use Case Map (UCM) (Giorgini *et al.*, 2005).

The benefit from using GRL is that it can be integrated with a scenario notation and can define a clear separation between its elements and their graphical representation. Moreover, it enables a scalable and consistent representation for multiple diagrams for the same goal model (Giorgini *et al.*, 2005).

It has been influenced by i* language in:

- GRL goal model has three basic concepts which are actors, intentional elements and links (Yu, 1997; Quartel *et al.*, 2009)
- GRL links types are decomposition, contribution, dependencies (Giorgini *et al.*, 2005)

Link Type		Original Rule
Dependency	i_d —D— i_s	$(v \in V)\ v(i_s) \Rightarrow v(i_d)$
Decomposition	i_d ; $i_1 \ldots i_n$	$(\bigwedge_{j=1}^{n} FS(i_j)) \Rightarrow FS(i_d)$ $\ldots$ $(\bigvee_{j=1}^{n} FD(i_j)) \Rightarrow FD(i_d)$
Contribution	i_s —C— i_d	$(c = \text{Make})\ FS(i_s) \Rightarrow FS(i_d)$ $(c = \text{Help})\ FS(i_s) \Rightarrow PS(i_d)$ $\ldots$ $(c = \text{Unk}, v \in V)\ v(i_s) \Rightarrow U(i_d)$ $(c \in \{\text{Make, Help} \ldots\})\ U(i_s) \Rightarrow U(i_d)$
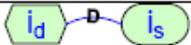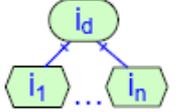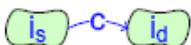
Fig. 3. i* forward propagation rules (Torkar *et al.*, 2009)

Goal   **Invariant Axioms**

$G:$ $FS(G) \to PS(G), \quad FD(G) \to PD(G)$

Goal relation   **Relation Axioms**

$(G_1, \ldots, G_i, \ldots G_n) \xrightarrow{and} G:$ $(\bigwedge_i FS(G_i)) \to FS(G),\ (\bigwedge_i PS(G_i)) \to PS(G)$

$\bigwedge_i (FD(G_i) \to FD(G)),\ \bigwedge_i (PD(G_i) \to PD(G))$

$(G_1, \ldots, G_i, \ldots G_n) \xrightarrow{or} G:$ $(\bigwedge_i FD(G_i)) \to FD(G),\ (\bigwedge_i PD(G_i)) \to PD(G)$

$\bigwedge_i (FS(G_i) \to FS(G)),\ \bigwedge_i (PS(G_i) \to PS(G))$

$G_2 \xrightarrow{+S} G_1:$ $PS(G_2) \to PS(G_1)$

$G_2 \xrightarrow{-S} G_1:$ $PS(G_2) \to PD(G_1)$

$G_2 \xrightarrow{++S} G_1:$ $FS(G_2) \to FS(G_1),\ PS(G_2) \to PS(G_1)$

$G_2 \xrightarrow{--S} G_1:$ $FS(G_2) \to FD(G_1),\ PS(G_2) \to PD(G_1)$

$G_2 \xrightarrow{+D} G_1:$ $PD(G_2) \to PD(G_1)$

$G_2 \xrightarrow{-D} G_1:$ $PD(G_2) \to PS(G_1)$

$G_2 \xrightarrow{++D} G_1:$ $FD(G_2) \to FD(G_1),\ PD(G_2) \to PD(G_1)$

$G_2 \xrightarrow{--D} G_1:$ $FD(G_2) \to FS(G_1),\ PD(G_2) \to PS(G_1)$

Fig. 4. Tropos forward propagation rules (Kiyavitskaya and Zannone, 2008)

The main differences respecting i* are that GRL offer a new link type namely, correlation link. Correlation link describes the side effects of an element on another element (Giorgini *et al.*, 2005). Also, GRL offers constructors for enabling relationships with external elements (Yu, 1997). Moreover, the use of URN links and metadata for enabling linking GRL with UCM elements. However, GRL supports only one type of actor, whereas i* support the notations of role, agent and position (Quartel *et al.*, 2009).

There exist many GRL tools, some of them are jUCMNav and OME (Yu, 1997; Quartel *et al.*, 2009). JUCMNav tool is an Eclipse plug-in for the creation, analysis and transformation of URN models (Quartel *et al.*, 2009; Giorgini *et al.*, 2005).

Amyot presented a tool providing a lightweight profile for GRL that enables creating a goal model in i* style (Quartel *et al.*, 2009). Such profile supplements GRL with i* missing concepts using the advantage of URN links and metadata. Moreover, it restricts the usage of GRL to i* through the usage of UML's Object Constraint Language (OCL). The tool is implemented in the jUCMNav tool.

The result from criterion 10 which is analysis of early requirements goal models is detailed and defined for i* framework, Tropos and GRL.

The goal satisfaction analysis procedure is applied on a goal model in order to select alternatives that aim to satisfy the desired goal. Goal satisfaction is either forward or backward propagation whether it is qualitative or quantitative (Yu, 1997; Torkar *et al.*, 2009; Massaccia *et al.*, 2005).

Forward Propagation

The forward propagation starts by initializing a set of alternative with a satisfaction value and then propagates such values upward iteratively through links and forward propagation rules until reaching the top goals (Amyot and Mussbacher, 2008).

Backward Propagation

The backward propagation starts by initializing the top desired goals with satisfaction values and then propagates such values downward iteratively through links and backward propagation rules (Amyot and Mussbacher, 2008):

- i* framework analysis supports both forward and backward propagation analysis on both qualitative and quantitative (Amyot and Mussbacher, 2008)
- Tropos analysis supports both forward and backward propagation analysis on both qualitative and quantitative. It solves the back propagation problem using two approaches. In the first approach, it reduces the problem of the backward propagation to that of the propositional Satisfiability (SAT). In the second approach, Tropos tries to find the set of alternatives with minimum cost that achieve the desired top goals using the Minimum-Weight Propositional Satisfiability (MW-SAT) (Kiyavitskaya and Zannone, 2008)
- GRL analysis supports both forward and backward propagation analysis on qualitative, quantitative and hybrid analysis (Massaccia *et al.*, 2005)

The results for criterion 11 which resulted in using the paper written by Torkar *et al.* (2009) was dealing with the challenge of making decisions after modeling early requirements with the presence of uncertainty for these requirements.

In real world applications, most of the goal models suffer from uncertainty and this is due to having gaps in the knowledge domain, disagreement between stakeholders or the presence of uncertainty over requirement details. Therefore, it is important to handle uncertainty since ignoring uncertainty may lead to selecting alternatives that may not be sufficient to achieve the desired goals or eliminating viable alternatives.

$$FS(G) \rightarrow \begin{cases} \bigwedge_i FS(G_i) & \vee \quad \text{If } (G_1,...,G_t,...G_n) \xrightarrow{and} G \\ \bigvee_i FS(G_i) & \vee \quad \text{If } (G_1,...,G_t,...G_n) \xrightarrow{or} G \\ FS(G_t) & \vee \quad \text{For every } R_i : G_t \xrightarrow{++S} G \\ FD(G_t) & \quad \text{For every } R_i : G_t \xrightarrow{--D} G \end{cases}$$

$$FD(G) \rightarrow \begin{cases} \bigvee_i FD(G_i) & \vee \quad \text{If } (G_1,...,G_t,...G_n) \xrightarrow{and} G \\ \bigwedge_i FD(G_i) & \vee \quad \text{If } (G_1,...,G_t,...G_n) \xrightarrow{or} G \\ FD(G_t) & \vee \quad \text{For every } R_i : G_t \xrightarrow{++D} G \\ FS(G_t) & \quad \text{For every } R_i : G_t \xrightarrow{--S} G \end{cases}$$

$$PS(G) \rightarrow \begin{cases} \bigwedge_i PS(G_i) & \vee \quad \text{If } (G_1,...,G_t,...G_n) \xrightarrow{and} G \\ \bigvee_i PS(G_i) & \vee \quad \text{If } (G_1,...,G_t,...G_n) \xrightarrow{or} G \\ PS(G_t) & \vee \quad \text{For every } R_i : G_t \xrightarrow{++S} G \\ PD(G_t) & \vee \quad \text{For every } R_i : G_t \xrightarrow{--D} G \\ PS(G_t) & \vee \quad \text{For every } R_i : G_t \xrightarrow{+S} G \\ PD(G_t) & \quad \text{For every } R_i : G_t \xrightarrow{-D} G \end{cases}$$

$$PD(G) \rightarrow \begin{cases} \bigvee_i PD(G_i) & \vee \quad \text{If } (G_1,...,G_t,...G_n) \xrightarrow{and} G \\ \bigwedge_i PD(G_i) & \vee \quad \text{If } (G_1,...,G_t,...G_n) \xrightarrow{or} G \\ PD(G_t) & \vee \quad \text{For every } R_i : G_t \xrightarrow{++D} G \\ PS(G_t) & \vee \quad \text{For every } R_i : G_t \xrightarrow{--S} G \\ PD(G_t) & \vee \quad \text{For every } R_i : G_t \xrightarrow{+D} G \\ PS(G_t) & \quad \text{For every } R_i : G_t \xrightarrow{-S} G \end{cases}$$

Fig. 5. Tropos backward propagation rule (Kiyavitskaya and Zannone, 2008)
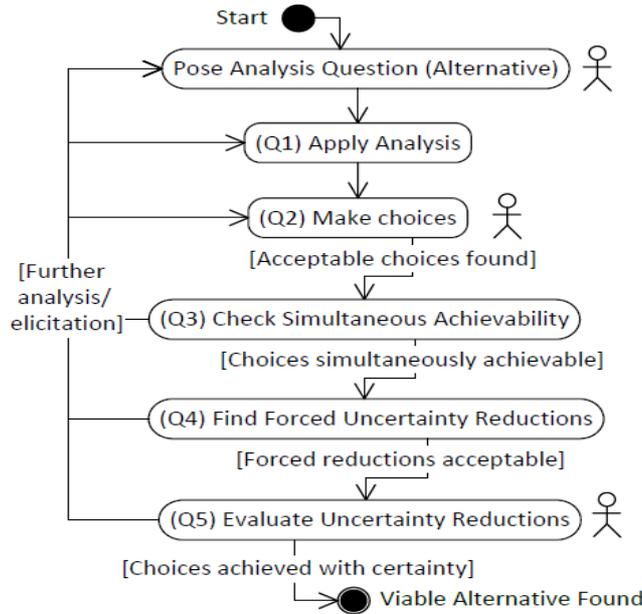
Fig. 6. Horkoff goal model analysis methodology (Torkar *et al.*, 2009)

Horkoff proposed a semi-automated tool (Torkar *et al.*, 2009) to handle uncertainty in early requirements represented in i* using MAVO framework in addition to her proposed formal analysis formula based on the forward qualitative analysis. Horkoff goal model analysis methodology is depicted in Fig. 6. Initially, it constructs a set of all possible concretizations where each of them represents a concrete goal model results from resolving a specific uncertainty requirement. Then, for each concretization she applies forward qualitative analysis. Afterwards the user is allowed to select her choices for each top goal which is then checked for simultaneous availability in any of the possible set of concretization. Reduced uncertainty respecting the user acceptability and the domain consistency is evaluated for the concretization results in the simultaneous achievable choices. Any appearing changes will lead to further analysis.

The results for contextual goal modeling and reasoning framework and that is criteria 12 defined that more than one alternative can satisfy the top desired goal. The decision of selecting among them depends on the applied context. Thus, it is important to enrich goal models with context. Consequently, Raian proposed a goal-oriented requirement engineering modeling and reasoning framework for systems operating under various contexts (Ghanavati *et al.*, 2009). The framework relates context and goals using Tropos.

Then, it analyzes all contexts to identify ways for verifying them. In order to derive requirements reflecting a certain context automatically, the framework proposes two reasoning techniques, namely, design time and runtime reasoning techniques. The runtime reasoning technique concerns deriving goal model variants reflecting context and user priorities. The design time reasoning technique concerns deriving requirements for the system-to-be with minimum cost and valid in all considered contexts.

Conclusion

In this study we presented a SLR concerning goal modeling techniques in RE. It has been realized that goal modeling techniques are being preferred than traditional methods such as object oriented approaches. Our search showed that requirements are considered to be a subset of goals and in return goals have been the major umbrella that concerns everyone for modeling. Moreover, we found that late requirements modeling haven't got so much interest such as that of early requirements modeling. We have discussed three well known early requirements models which are: i* framework, Tropos and GRL. Also, we have addressed in brief UML which is considered the most popular late requirements modeling technique. The below table also briefs the i* model and its variants as follow:

Comparison criteria	i*	Tropos	GRL
RE life cycle	Early phase	Early, late requirement. Architectural, detailed design	Early phase
Type of models	SR, SD	Actor diagram, Rationale diagram	None
Types of actors	Agent, position, role	Agent, position, role	Actor
Intentional elements	Goal, soft goal, task, resource	Goal, soft goal, task, resource	Goal, soft goal, task, resource
Relationship among actor	-Relationships among actors by means of intentional elements: "dependency link" -Relationships among specific types of actors: "occupies", "plays", "covers" -Relationship among some actor type: "is-part-of"	-Relationships among actors by means of intentional elements: "dependency link" -Relationships among actor types: "occupies", "plays", "covers"	-Relationships among actors by means of intentional elements: "dependency link"
Relationship among external elements	-Task-decomposition (using AND) -Means-end (using OR) -Contribution link	-AND/OR decomposition -Means-ends(using OR) -Contribution link	-Decomposition -Means-ends -Contribution link -Correlation link
Analysis procedure	Forward, backward qualitative	Forward, backward qualitative and quantitative	Forward, backward qualitative, quantitative and hybrid
Tools	OME, REDEPEND	GR-Tool	jUCMNav, OME
Handle context	None	Using contextual goal modelling framework	None
Handle uncertainty	Using MAVO		
Uncertainty types	None	None	
Security	Supports secure i* language (SI*).	Supports secure Tropos	None

Author's Contributions

Iman Ahmed ElSayed: Equal contribution with Zeinab Ezz in preparing, writing the research and collecting data.

Zeinab Ezz: Equal contribution with Iman Ahmed in preparing, writing the research and collecting data

Eman Nasr: Mentor for Iman Ahmed and Zeinab Ezz for their contribution

Ethics

This article is original and contains unpublished material. The corresponding author confirms that all of the other authors have read and approved the manuscript and no ethical issues involved.

References

Amyot, D. and G. Mussbacher, 2008. Development of telecommunications standards and services with the user requirements notation. Proceedings of the Workshop on ITU System Design Languages, (SDL' 08), Geneva, Switzerland, pp: 15-16.

Amyot, D., J. Horkoff, D. Gross and G. Mussbacher, 2009. A lightweight GRL profile for i* modeling. Proceedings of the Workshops on Advances in Conceptual Modeling, Nov. 9-12, Springer, Gramado, Brazil, pp: 254-264. DOI: 10.1007/978-3-642-04947-7_31

Ayala, C.P., C. Cares, J.P. Carvallo, G. Grau and M. Haya *et al.*, 2005. A comparative analysis of i*-based agent-oriented modeling languages. Proceedings of the 7th International Conference on Software Engineering and Knowledge Engineering, KSI Press, Taipei, pp: 43-50.

Franch, X., L. López, C. Cares and D. Colomer, 2016. The *i* Framework for Goal-Oriented Modeling. In: Domain Specific Conceptual Modeling, Karagiannis, D., H.C. Mayr and J. Mylopoulos (Eds.), Springer, pp: 485-506.

Ghanavati, S., D. Amyot and L. Peyton, 2009. Compliance analysis based on a goal-oriented requirement language evaluation methodology. Proceedings of the 17th IEEE International Requirements Engineering Conference, Aug. 31-Sept. 4, IEEE Xplore Press, pp: 133-142. DOI: 10.1109/RE.2009.42

Giorgini, P., J. Mylopoulos and R. Sebastiani, 2005. Goal-oriented requirements analysis and reasoning in the tropos methodology. J. Eng. Applic. Artificial Intell., 18: 159-171. DOI: 10.1016/j.engappai.2004.11.017

Horkoff, J. and E. Yu, 2016. Interactive goal model analysis for early requirements engineering. Requirements Eng., 21: 29-61. DOI: 10.1007/s00766-014-0209-8

Horkoff, J., R. Salay, M. Chechik and D. Sandro, 2014. Supporting early decision-making in the presence of uncertainty. Proceedings of the IEEE 22nd International Requirements Engineering Conference, Aug. 25-29, IEEE Xplore Press, pp: 33-42. DOI: 10.1109/RE.2014.6912245

Kavakli, E. and P. Loucopoulos, 2003. Goal driven requirements engineering: Evaluation of current methods. Proceedings of the 8th International Workshop on Evaluation of Modeling Methods in Systems Analysis and Design, (SAD' 03), Velden, Austria.

Kiyavitskaya, N. and N. Zannone, 2008. Requirements model generation to support requirements elicitation: The Secure Tropos experience. Automated Software Eng., 15: 149-173. DOI: 10.1007/s10515-008-0028-6

Massaccia, F., M. Prest and N. Zannone, 2005. Using a security requirements engineering methodology in practice: The compliance with the Italian data protection legislation. Comput. Standards Interfaces, 27: 445-455. DOI: 10.1016/j.csi.2005.01.003

Quartel, D., W. Engelsman, H. Jonkers and M.V. Sinderen, 2009. A goal-oriented requirements modelling language for enterprise architecture. Proceedings of the IEEE International Enterprise Distributed Object Computing Conference, Sept. 1-4, IEEE Xplore Press, pp: 3-13. DOI: 10.1109/EDOC.2009.22

Torkar, R., T. Gorschek, R. Feldt, U.A. Raja and K. Kamran, 2009. Requirements traceability state-of-the-art: A systematic review and industry case study. IST J.

VAn Lamsweerde, A., 2001. Goal-oriented requirements engineering: A guided tour. Proceedings of the 5th IEEE International Symposium on Requirements Engineering, Aug. 27-31, IEEE Xplore Press, pp: 249-262. DOI: 10.1109/ISRE.2001.948567

Yu, E.S., 1997. Towards modelling and reasoning support for early-phase requirements engineering. Proceedings of the 3rd IEEE International Symposium on Requirements Engineering, Jan. 6-10, IEEE Xplore Press, pp: 226-235. DOI: 10.1109/ISRE.1997.566873