# Eliciting Multi-Agent Systems Intentionality: from Language Extended Lexicon to i* Models

Antonio de Padua Albuquerque Oliveira
*UERJ (Brasil); PUC-Rio (Brasil)*
*padua@ime.uerj.br ; padua@inf.puc-rio.br*

Julio Cesar Sampaio do Prado Leite
*PUC-Rio (Brasil)*
*www.inf.puc-rio.br/~julio*

Luiz Marcio Cysneiros
*York University (Canada)*
cysneiro@yorku.ca

Claudia Cappelli
*PUC-Rio (Brasil)*
ccappelli@inf.puc-rio.br

## Abstract

*MAS methods still lack coverage to the goal elicitation process, especially on how to identify goals from corporate information, mission statements and from interviews with stakeholders. Only after eliciting goals we will be able to properly deal with goal models. On the other hand, intentionality models, for example the i* Framework, are, usually, complex and difficult to read. By contrast, this paper proposes an indirect inquire process that can identify goals in a bottom-up and simple elicitation approach together with a proposal to reduce the problem of scalability of i* models.*

## 1. Introduction

When talking about intentionality we are directed to consider a GOAL-ORIENTED APPROACH and therefore, we need to understand and define WHY we are using goal modeling. The goal concept has come to play a critical role in Requirements Engineering (RE). In RE, goals are considered a significant construct. The importance researchers give to goals can be seen in the several approaches proposed which use goals as central to their methods, models and tools: KAOS [1], i* Framework [2], GBRAM [3], Tropos [4] [5], and NFR Framework [6] are well cited exemplars. A. van Lamsweerde classifies them as belonging to GORE (Goal-Oriented Requirements Engineering). Various researchers consider GORE one of the best ways to produce quality software and therefore because MAS deals with agents' goals, commitments, beliefs, and abilities [27], the GORE approach seems particularly applicable to MAS.

We face a common misuse, in the software engineering community, of the goal concept. Many people believe, wrongly, that a goal is like a function or an action that stakeholders can perform. It is frequent to see system's goals like: "Update clients' information", "Edit material invoices of suppliers", and "Print the accounts before the due date". Although those may be goals in a lower level of abstraction they are not in fact the goals driving the business, which are key to be elicited if we desire a proper RE process.

Kavakli and Loucopoulos [7] have demonstrated that goals have been applied within RE with a different aim or purpose and one big problem with goals use occurs in the way they are communicated (their forms). Goals cover two important concerns: nature and representation. Goals are abstract by nature [8] and hence it may be hard for many people to understand the meaning and the power of GORE approach.

Our proposal uses goals and softgoals in the same way used by the i* Framework [2], [9]. In order avoid free style representations, which allow a goal to be represented like a function or an action; we adopted pre-defined frames that have the purpose of driving the requirements engineers to represent stockholder's intentionality.

One important gap in GORE approaches is the fact that all those methods do not deal specifically with intentionality elicitation. All of them, no exception, are strongly oriented towards modeling. The same happens with the BPM (Business Process Management) area

[26]. Although some BPM methods consider goals elicitation BPM does not consider strategic dependencies.

Another motivation for this work relies on the common misuses of i* models pointed out by Estrada [10] and Pastor [11]. Ideally i* models should be divided into small pieces avoiding scalability problems and also improving the stakeholders' understanding.

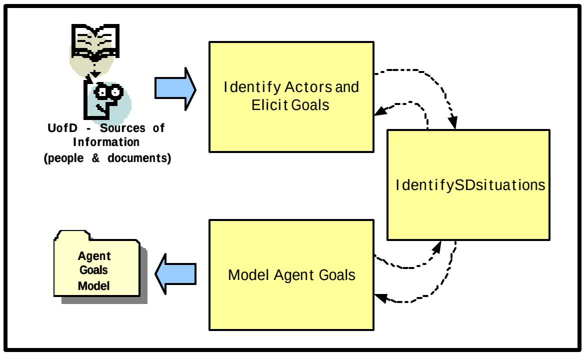



Figure 1 – Overview of the Proposed Method AGFL. (AGFL – Agent Goals from Lexicon)

In this work we introduce the concept of AGFL – Agent Goals from Lexicon showed in Figure 1, which is formed by tree steps. In the first step ***"Identify Actors and Elicit Goals"*** the engineer captures goals (and softgoals), separates them by actors and organizes them in a chronological order. In the second step ***"Identify SDsituations"*** the engineer identifies goals (and softgoals) arrangements that are connected in order to implement situations of dependency called SDsituations – Strategic Dependency Situations [12]. In the last one ***"Model Agent Goals"*** the requirements engineer builds diagrams, a kind of state charts that considers actors/agents, in order to represent chains of goals (and softgoals) relationships. The diagrams are called "INTENTIONALITY PANELS" and they are a simpler view of the i* Framework SR model.

For pushing up AGFL intentionality elicitation, the proposed method chose the Language Extended Lexicon (LEL) [13] because LEL promotes the capture of hints to find goals. As LEL captures the application vocabulary elements and classifies them as: subject, (someone who does the action) object, (something that receives the action) verb, (that means the action) and state (that is a result of the action), it provides a proper base to find application goals.

AGFL method chose the i* Framework [2] for requirements modeling, first because i* is intentionality based, second because i* recognition: two other methods are i* based (GRL [25] and TROPOS [4], [5]) and, beyond that, because i* deals with intentionality in a complete way, using the idea of actors strategic dependency, and more i* is one of the most cited and discussed requirements modeling framework.

The paper is organized as follows: Section 2 presents LEL - Language Extend Lexicon, the i* Framework and SDsituations Technique as well. We detail the steps of our method at Section 3. In Section 4 we illustrate the use of our method. And, Section 5 concludes stressing the novelty of our work and points out to future research.

## 2. LEL, the i*, and SDsituations

### 2.1. Language Extend Lexicon Approach

Language Extend Lexicon (LEL) [13] is a representation model of terms used in the application language. LEL implements a very simple idea: understand the language used by the application without worrying about understanding the application. Thus, the aim of the LEL is to represent words or sentences (called symbols) peculiar to the Universe of Discourse – UofD[1]. Requirements Elicitation Techniques [14] [15] must be applied in order to obtain the LEL.

LEL is composed by symbols. Each symbol, or entry, is identified by a name or names (case of synonyms) and is represented by two descriptions. The first one, called notions, is the **denotation** of the symbol, equivalent to a description found in a dictionary. The second one, called behavioral response, is the **connotation** of the symbol, which describes the contextualization of that symbol in the UofD (e.g.: What actions are influenced by the symbol?). Moreover, both notion and behavioral responses use other existing symbols to describe a new symbol. The symbols in LEL are classified into four different kinds: state, verb, object, and subject. Symbols representation must follow (1) the circularity principle (also called "closure principle") and (2) the minimal vocabulary principle. The circularity principle states that we have to maximize the use of LEL symbols when describing a symbol while the minimal vocabulary principle states that we have to minimize the use of words that are external to the Lexicon. These two principles are significantly important for LEL to be self-contained and highly

[1] UofD [13] means the overall context in which the software will be developed and operated. The UofD includes all the sources of information and all the people related to the software.

coupled [13].

This activity is supported by the C&L tool software, which is a management tool for lexicons and scenarios. C&L is an open tool developed by the Requirements Engineering Group at PUC-Rio and is available at http://pes.inf.puc-rio.br/cel/.

## 2.2. The i* Framework

I* modeling framework [2] models organizational contexts based on the dependency relationships among actors. The central idea of i* is that: actors depend on each other for goals to be achieved, for resources to be provided, for tasks to be performed, and for softgoals to be satisficed[2]. I* Framework deals with two kinds of models: the Strategic Dependency (SD) model and the Strategic Rationale (SR) model.

External relationships among actors are expressed in the Strategic Dependency (SD) model. The SD model depicts the organizational environment context of the system as a network of dependency relationships among actors. This network consists of a set of nodes and links where each node represents an actor and each link maps out one dependency between two actors. The Strategic Rationale (SR) model expresses organizational context and also internal relationships among the intentional elements within an actor's reasoning. Rationales are modeled through means-ends relationships, task decompositions, and softgoal contributions.

## 2.3. Strategic Dependency Situations - SDsituations

Situations of dependency occur in the organizational environment and the essential idea of SDsituations is: "each dependency link (goal, softgoal, task or resource) that involves two actors is not isolated"; it is part of one well distinct situation of collaboration called one "Strategic Dependency situation" or one SDsituation [12]. One SDsituation is composed by one or more dependency elements and any SDsituation can be identified separately from another SDsituations. For example: An electoral process can only be performed if the candidates were defined previously. It means that we should identify two separate situations but one depends on the other: The **election process** situation depends on **candidates definition** situation. An SDsituation can be characterized as a business unity part. It means that we should identify several separate SDsituations but one depends on the others, forming a chain of interdependencies. Interdependencies among SDsituations may be physical, logical or temporal and can be represented in a particular diagram [12]. Figure 2 shows an SDsituation illustration.

[2] "Softgoals are satisficed, rather than satisfied" [6].

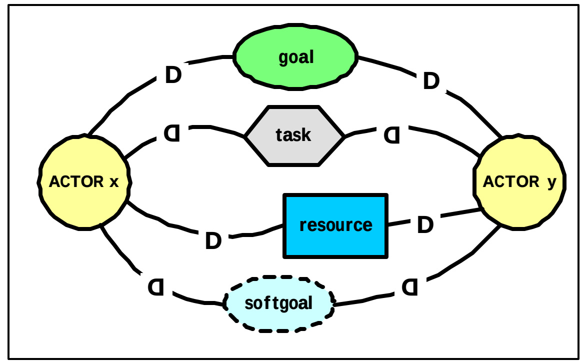



Figure 2 – Illustration of an SDsituation. Note that an SDsituation can be composed by four kinds of elements with zero or more elements of each kind.

## 3. The Elicitation Method: AGFL - Agent Goals from Lexicon

Our proposal aims at eliciting the intentionality. To accomplish that, we need to identify the owners of the goals and softgoals i.e., the social actors in the organization. We assume that actor identification is doable, since they are frequently directly mentioned in documents and interviews. The kernel of the problem is how to elicit and express intentionality. Intentionality means actors' interests and motivations [16] and it is the central goal of this work. We introduce a technique for systematically capture the intentionality of social actors before modeling.

Our proposition is an indirect process of inquiry based [17] on LEL. Note that inside each LEL symbol (see two examples in Figure 5) the behavioral responses aim at representing actions: which actions happen, which actions are the reflexes of an action, which actions can be applied and about actions that can occur. Our idea is simple: "ACTIONS CHANGE STATES AND STATES ARE GOALS". "A goal is a condition or state of affairs in the world that the actor would like to achieve" Eric Yu [2].

In order to discover the state of affairs that one action wants to change we need to ask "why?" the action occurs.

Goals (and softgoals), which should be the main source of requirements, have two connotations inside of our approach: form and substance. Form means the appearance (conformation) "how goals are communicated" and substance (content) is associated with significance "what consequence a goal has". In order to follow this idea we adopted frames for guiding requirements engineer. We created three frames stimulated by the four kinds of LEL's symbols. The frames were prepared to receive short answers that the behavioral responses of LEL's symbols are mentioning. Each frame receives "the answers for why questions".

Therefore, in our approach goals can have a structured syntax; (object + BE + verb in passive). We fixed some parts in order to facilitate the requirements engineer's work. Goals should be described based on LEL symbols and also based on a list containing the most common domain verbs (in the past tense), which are normally repetitive. This aims at facilitating to express the right syntax of the requirements engineer's work.

Figure 1 is an overview of the method. In this section we present heuristics using concepts, steps, activities, and helpful hints. An example of the method is shown in Section 4.

### *Identify Actors and Elicit Goals*

-- Prepare the LEL for the application domain:

Use all sources of information you need from the UofD. Define symbols into the LEL and classify them as: subject, object, verb, and state. Follow instructions for name, notion, and behavioral response definitions. Give attention to the principles of: circularity and minimum vocabulary.

-- Identify actors from LEL symbols:

Subject kind symbols are natural candidates to be actors.

-- Extract goals from symbol's behavioral responses:

Use the specific "frame" (see Frames 6a - subject, 6b - object, and 6c – verb and state) for each kind of LEL symbol. Each behavioral response must generate a goal into the pre-defined format answering the simple question "WHY".

If the answer builds a goal with a "dependency" for another actor, you must generate another goal (it is a reflexive one) despite of the dependency. A behavioral response "why" answer may indicate that the first actor (the depender) either "depends on", or needs help of another actor (the dependee) for his goal to be achieved. This kind of occurrence points to a "reflexive goal" which must be defined in addition.

In the sentence of goal definition always use the symbol's main name, do not use a synonym. Create goals in straight way sentences. For example: Instead of "rental payment be resolved" we may use "rental be paid", because a direct communication is more easier to be understood, as well as instead of "covered vehicle be repaired" we should use "vehicle be repaired", because only covered vehicles can be repaired.

-- Refine goals:

Group goals (type: subject) by actor. Put together goals that came from agents/positions/roles and belong to the same actor.

-- Convert goals from "object kind" to "subject kind". Because object kind frame has different shape, goals must be converted to the main shape.

-- Put together goals (verb and state kind) by actor.

-- Delete repeated or redundant goals. Give a temporal order for goals. Let long-term goals at the end of the set. Long-term goals are the ones that point to general or business results.

### *Identify SDsituations*

-- Distinguish SDsituations:

Determine sets of goals which should be separated in one business situation; goals may belong to one or more actors and they are strongly connected in a way to conclude a situation. The Requirements Engineer should note that there is no goal with weak connectivity (why?). In the same SDsituation, goals connectivity should not be temporal. Into SDsituations actors can be specialized in agents, roles and positions [18], [19].

-- Recognize interdependencies among SDsituations: Observe each SDsituation and recognize situations that have logical, temporal or sequential dependency with another situation. It means that we should identify two separate situations but one depends on the other critically. Detect, also, whether there is some parallelism among SDsituations.

-- Create the SDsituations diagram:

Show all SDsituations in the same diagram and show the interdependencies among them. Illustrate the factor time in the diagram, when necessary.

### *Model Agent Goals*

-- Create the Intentionality Panels:

Prepare one panel for each SDsituation. Decide about the order of goals and softgoals in the actor's axis. Goals or softgoals that need to wait for some "state" should be placed in the upper place. Represent the relationships among goals and softgoals. Prefer to set closer actors who have more numbers of dependencies.

-- Model goals and softgoals effects:

Since a complete SR model has the tendency of being huge and hard to follow, [11] and [21]; our method suggests that intentionality should be drawn in parts based in SDsituations [12] in a new diagram, called "Intentionality Panel" – IP diagram. This diagram is a reduction of the SR model, it considers the i* "means-ends" structure being represented only by the structure end (goal or the softgoal) and the relationships between goals and softgoals are thus represented. An "Intentionality Panel" – IP diagram, is a kind of state-chart diagram which also represents actors/agents. It is a kind of state-chart because it has different states linked together in a chain actors/agents' goals and softgoals.

In one IP diagram goals and softgoals are the elements (nodes) and there are four main types of relationships (links) among goals and softgoals:

a) **<u>Correlation</u>** *from goal$^2$ to goal$^1$:* goal$^1$ has goal$^2$ being a sub-component (goal$^2$ is a task component of goal$^1$). **<u>Correlation</u>** *from goal to softgoal:* the softgoal has the goal being a sub-component (the goal is a task component in the means-ends of the softgoal). **<u>Correlation</u>** *from softgoal to goal:* it means that the softgoal is a task component of the goal.

b) **<u>Contribution</u>** *from softgoal to softgoal:* it means that one softgoal can give a positive (+), neutral (?) or negative (-) contribution for another softgoal, exactly with the same semantic used in SR models.

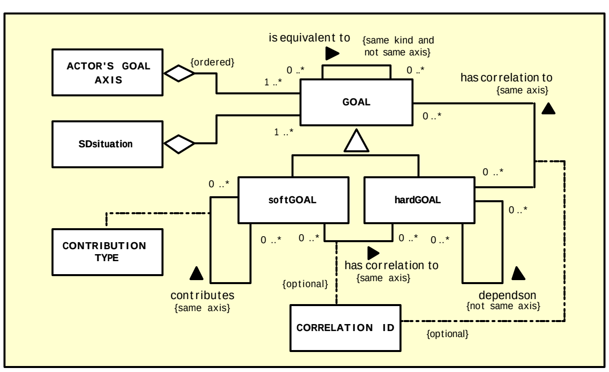



Figure 3 – Intentionality Panel class diagram. Note that two different kinds of goals can be represented in the Intentionality Panel.

c) **<u>Dependency</u>** *from goal$^4$ to goal$^3$*: goal$^3$ depends on goal$^4$ either for a task, a resource, a softgoal or a goal, exactly with the same semantic used for actors in SD models, but the element of dependency is not important so far.

d) **<u>Equivalence</u>** *of goals or softgoals:* it means that actors' softgoal or goal has the same importance but they do not have the same "mean" (task or implementation).

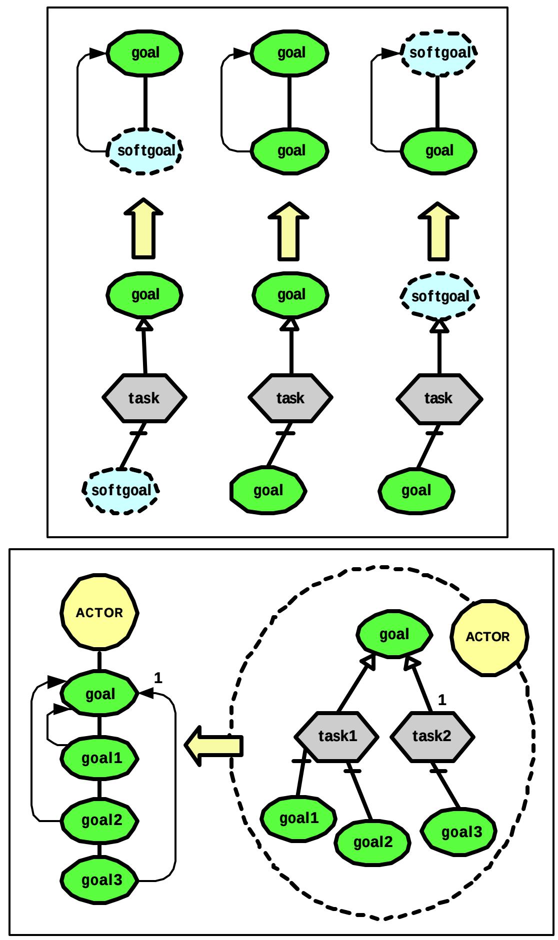



Figure 4 – In the upper part are represented the correspondences between SR model and the Intentionality Panel in the representation of **correlation**. In the lower part alternatives are represented: By one hand, goal$^1$ and goal$^2$ together they have correlation to the main goal be achieved but on the other hand, with ID=1, goal$^3$ has an alternative correlation to the main goal to be achieved.

Figure 3 shows "Intentionality Panel" elements relationships using a class diagram. Note that "goals" are organized into SDsituations because a set of interconnected goals defines one SDsituation. Actor's goals may have correlation among themselves and a goal correlation may have an ID, when it is necessary to show alternatives. When a softgoal contributes to another softgoal it is necessary to indicate the contribution type **(+, ?, -)**. Strategic dependencies between actors are represented by "depends on" association. Although actor's goals can be equivalent, they have different operationalizations.

Figure 4 shows how the i* SR model can be reduced into one "Intentionality Panel" – IP diagram, all kinds of relationships between goals and softgoals are illustrated.

## 4. Example of Eliciting Intentionality – The Insurance Company problem

### *Identify Actors and Elicit Goals*

-- Prepare the LEL for the application domain:

Three sources of information were available for building this example: the first one was the insurance company problem definition in the book of Hammer & Champy [22], the second one was a "Car Owner Manual" from a Brazilian insurance company, and the third one was "Geico Insurance Terms": http://www.geico.com/about/InsuranceTerms.htm.

-- Identify actors from LEL symbols:

The identified actors were: APPRAISER, BODY SHOP, AUTHORIZED BODY SHOP, CAR OWNER, COMPANY, INJURIED PERSON, INSURANCE AGENT, PHYSICIAN, POLICE, RENTAL CAR, and WITNESS.

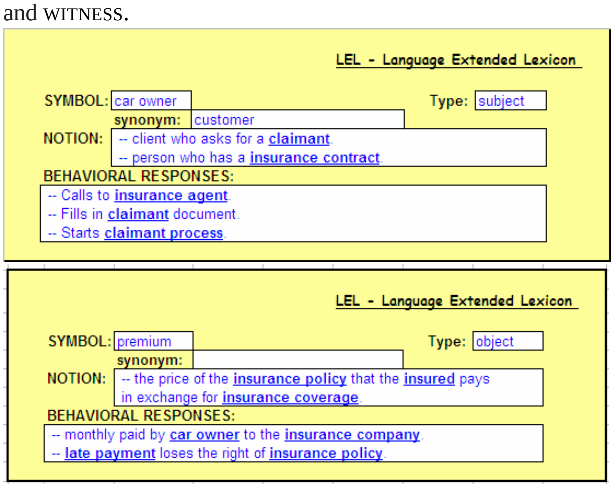


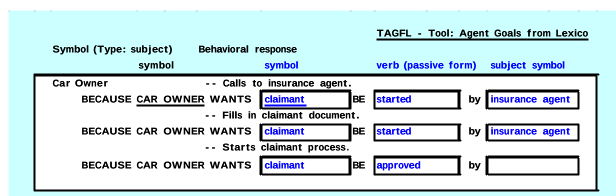

Figure 5 – Examples of LEL's symbols, one is a subject kind and the second is an object kind. Observe that others mentioned symbols are underlined in order to facilitate the circularity principle.

--Extract goals from symbol's behavioral responses:

Frame 6a shows three behavioral responses from symbol CAR OWNER and the answers for "why question". For example: "Why does the car owner **call the insurance agent**?"

As said in Section 3, when there is a second actor in the answer we have to answer another question. This new question is the case of a reflexive goal caused by a dependency. "Why does the insurance agent do this for the CAR OWNER?" and "Why does the COMPANY do this for the INSURANCE AGENT?" and also "Why does the APPRAISER do this for the COMPANY?" The frame below shows the answers.

Figure 6a shows the resulting goals created by Frame 6a. Because CAR OWNER depends on INSURANCE AGENT for "claimant BE started" we should know "why". The answer was filled in Frame 6a. And, because INSURANCE AGENT wants claimant BE approved by COMPANY we should know "why". By using this idea the requirement engineer must elicit the goals chain until attain a goal that does not depend on another actor.

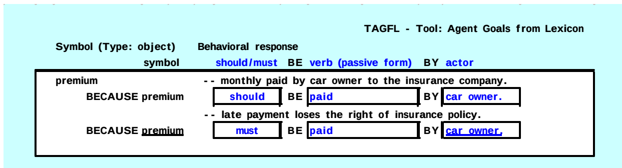



Frame 6a – Example of AGFL frame subject type for LEL's symbols. Note that requirements engineer, for answer the "why question", must fill one object, one verb and one subject for each behavioral response.

| | | | | | |
|---|---|---|---|---|---|
| CAR OWNER | | | | | |
| -- Calls to insurance agent | | | | | |
| Because car owner wants | claimant | BE | started | by | insurance agent |
| Because insurance agent wants | claimant | BE | approved | by | company |
| Because insurance agent wants | insurance | BE | renewed | by | |
| Because company wants | claimant | BE | validated | by | appraiser |
| Because appraiser wants | claimant | BE | approved | by | |
| -- Fills in claimant document | | | | | |
| Because car owner wants | claimant | BE | started | by | insurance agent |
| -- Starts claimant process | | | | | |
| Because car owner wants | claimant | BE | approved | by | |

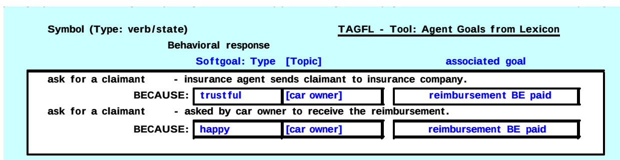
Figure 6a – Goals created in Frame 6a. There are two goals (second and third) created despite the insurance agent being a depender. Because second behavioral response goals chain is the same of the first one, it was omitted.

Frame 6b shows behavioral responses and goals revealed from object kind symbol premium and the answers for "why question". For example: "Why premium is monthly **paid by the car owner to the insurance company**?"




Frame 6b – Example of AGFL frame object type. Note that requirements engineer, to answer the "why question", must fill one verb and one subject for each behavioral response.

Frame 6c shows two more behavioral responses and goals revealed from the symbol (verb kind) <u>ask for a claimant</u>. Note the answers of "why question": "Why does insurance agent **send claimant to the insurance company?**" "Why does car owner **ask to receive the reimbursement?**" The reader also should note that each softgoal must be associated to a previous goal.




Frame 6c – Example of AGFL frame verb type. Note that requirements engineer, to answer the "why question", must fill softgoal type, must choose a topic in LEL and must choose an AGFL goal to be associated with.

Figure 6c.1 shows examples of softgoals discovered by LEL's symbols (verb kind). Figure 6c.2 shows softgoals discovered by LEL's symbols of the state kind.

| | | | | |
|---|---|---|---|---|
| Because | trustful | [customer] | police BE contracted | insurance agent |
| Because | trustful | [car owner] | reimbursement BE decided | company |
| Because | happy | [car owner] | reimbursement BE paid | company |
| Because | low cost | [reimbursement] | reimbursement BE paid | company |
| Because | low cost | [reimbursement] | reimbursement BE decided | company |

Figure 6c.1 – Softgoals created applying AGFL frame verb / state type. Note that softgoal's owner must be indicated.

| | | | |
|---|---|---|---|
| Because | customer | wants | valid [contract] |
| Because | insurance agent | wants | happy [car owner] |

Figure 6c.2 – Softgoals created by applying AGFL frame verb / state type. Note that goal's owner should be indicated automatically.

-- Refine goals:

The requirements engineer should separate goals by actors. Figure 7 shows car owner`s goals and dependency goals.

| | | | | | |
|---|---|---|---|---|---|
| Because | premium | BE | paid | by | car owner |
| Because car owner wants | claimant | BE | started | by | insurance agent |
| Because car owner wants | reimbursement | BE | decided | by | company |
| Because car owner wants | car | BE | repaired | by | body shop |

Figure 7 – CAR OWNER`s goals and dependency goals.

-- Convert goals from "object kind" (1) to "subject kind" (2). Figure 8 shows an example, Goal (1) came from previous activity and should be converted.

-- Put actor's goals together.

| | | | | | | |
|---|---|---|---|---|---|---|
| (1) | Because | premium | BE | paid | by | car owner |
| (2) | Because car owner wants | premium | BE | paid | | |

Figure 8 – CAR OWNER's object goal is converted to subject

-- Put together goals (verb and state kind) by actors.

| | | | | | |
|---|---|---|---|---|---|
| Because car owner wants | police | BE | contracted | | |
| Because car owner wants | premium | BE | paid | | |
| Because car owner wants | claimant | BE | started | by | insurance agent |
| Because car owner wants | deductible | BE | paid | by | |
| Because car owner wants | claimant | BE | approved | by | insurance agent |
| Because car owner wants | reimbursement | BE | decided | by | company |
| Because car owner wants | car | BE | repaired | by | body shop |
| Because car owner wants | repair | BE | executed | by | authorized body shop |

| | | | | |
|---|---|---|---|---|
| Because | trustful | [customer] | police BE contracted | insurance agent |
| Because | happy | [car owner] | reimbursement BE paid | company |

Figure 9 – CAR OWNER's goals and softgoals.

Figure 9 shows the elicited goals and softgoals related to CUSTOMER and CAR OWNER which is a specialized CUSTOMER.

### *Identify SDsituations*

-- Distinguish SDsituations:

In the Insurance Company example, we recognized 9 (nine) situations well distinguished, see Figure 10. For example: "Claimant Approval" is a business situation that has interconnected goals from actors: CAR OWNER, APPRAISER, and INSURANCE COMPANY. "Claimant Approval" can only be performed if the "Insurance Contract" had been previously sold. It means that we should identify two separate situations.

-- Recognize interdependencies among SDsituations:

SDsituations are logically, sequentially or temporally linked. We discovered 3 different types of "Repair Accomplishments" but "Rental Delivery Assistance" must occur only while vehicle was being repaired. And, the “Repair Process” must occur after the “Claimant Approval” and before the “Reimbursement Process”.

-- Create the SDsituations diagram:

Figure 10 shows an SDsituation diagram. A benefit of this model is that it can show the chain of strategic dependency situations.

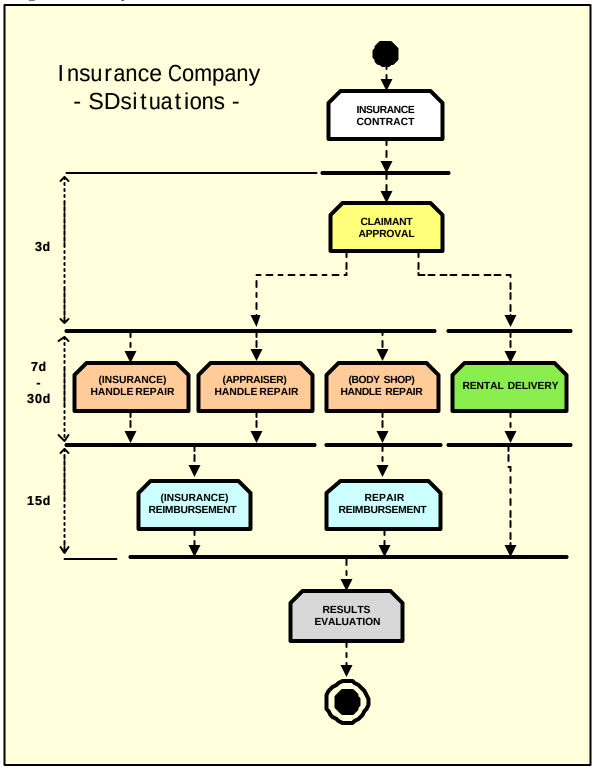



Figure 10 - The SDsituations Diagram.

### *Model Agent Goals*

-- Create the Intentionality Panels:

Figure 11 shows an Intentionality Panel diagram that mapped only one SDsituations”. A benefit of this model is that the modeler can concentrate in only one piece of draw which he needs to understand. The idea is to try as much as possible not to push human limits [23].

-- Model goals and softgoals effects:

SDsituation should be analyzed about goals effects; the idea is to show in the “Intentionality Panels” which goals are affected by a non achievement of one goal. In the Figure 11, the mark “V” means goal achieved and “X” means goal denied. The requirements engineer should analyze the effects if he wants in a more complete Intentionality Panel - IP diagram, involving whether all or almost all SDsituations and analyze problems in a whole IP.

By analyzing “intentional compromises” involving goals in the Intentionality Panel - IP diagram, one can discover some points of compromise that must be reconsidered for the system to be. For example: it should be better to have more than one “Rental Car” in order not to fail “client satisfaction”.

In the appendix we show as an example one SD model and one SR model, both based in one SDsituation: (APPRAISER) HANDLE REPAIR.

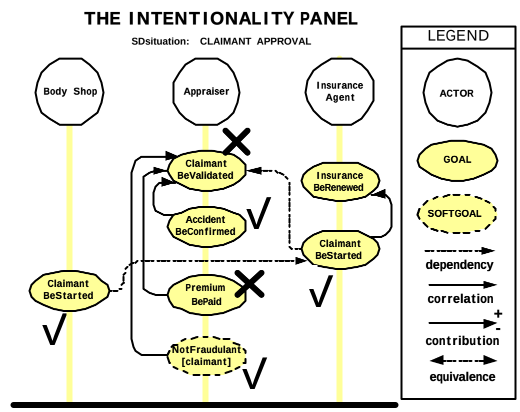



Figure 11 – Illustrations of one SDsituations IP diagram. Note that the color means that “goals and softgoals” belong to same SDsituation.

## 5. Conclusion

The proposed method, AGFL, brings goal elicitation as the prime concern, towards properly supporting MAS development.

The main contribution is to elicit agent goals by a method based on the Language Extend Lexicon [13] of the domain, which follows the simple idea represented bellow: “actions point to goals”.

-----------------------------------------------------------------

**LEL SYMBOLS’** ➔ are related with ➔ **ACTIONS**
**ACTIONS** ➔ have the ability of changing ➔ **STATES**
**STATES** ➔ are ➔ **GOALS**

-----------------------------------------------------------------

Therefore, our work created two structured diagrams. The SDsituations diagram has the aim of organizing agent goals in “Strategic Dependency situations” suggesting that SD and SR models should be divided considering the idea of SDsituations. The Intentionality Panel - IP diagram, has the aim of “making agent goals dependency explicit”, keeping the same semantic created by the i* Framework [2] and preparing the terrain for the requirements engineer for the decisions about dependency among elements. Moreover, stakeholders are more prone to understand SDsituations because SDsituations have a business perspective. We also sustain that SDsituations are a proper mechanism for scalability control by providing modularity constructs. We understand that the results we have achieved are also valid for traditional software systems since they can be seen as a particular case of MAS.

Based in the method, we intend to implement a software tool supporting traceability [24] and the baseline for requirements evolution [20]. The baseline traceability support should allow the process **forward**: *UofD* → *LEL* → *SDsituations* → *IP diagram* and **backward**: *IP diagram* → *SDsituations* → *LEL* → *UofD*.

We have applied the AGFL method for the Insurance Company case study as a proof of concept. Our results are encouraging; however, more research in the use of the AGFL is necessary. We need to apply the strategy in different situations in order to get practical evidence of the benefits of applying the approach in real cases. While carrying out these experiments we will also evaluate how well the approach scales to more complex problems.

**Acknowledgements:** *Thanks to* Bruno Gonzalez-Baixauli *for his valuable comments on this paper.*

# Appendix

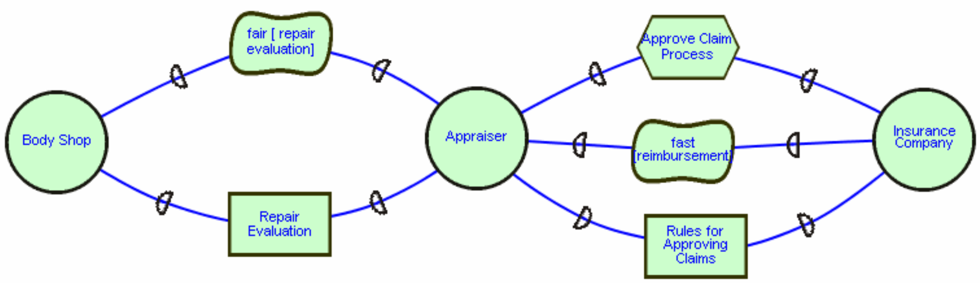



Figure 12 – SD model - SDsituation: (APPRAISER) HANDLE REPAIR

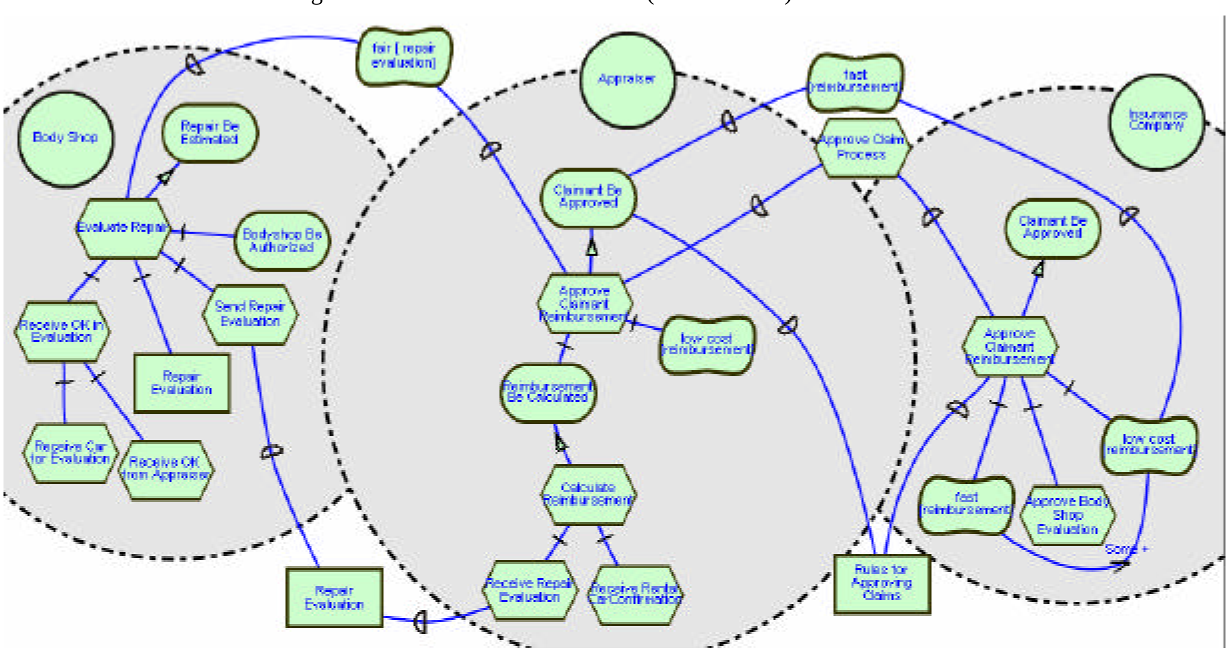



Figure 13 – SR model - SDsituation: (APPRAISER) HANDLE REPAIR

## 6. References


[1] Lamsweerde; Axel van; Goal-Oriented Requirements Engineering: A Guided Tour; Proceedings RE'01, 5th IEEE International Symposium on Requirements Engineering; Toronto; August 2001; 249-263.

[2] Yu, E. Modeling Strategic Relationships for Process Reengineering. PhD Thesis, Graduate Department of Computer Science, University of Toronto, Toronto, Canada, 1995, pp. 124.

[3] Antón, A.I. Goal Identification and Refinement in the Specification of Software-Based Information Systems, Ph.D. Thesis, Georgia Institute of Technology, 1997

[4] Castro, J.; Kolp, M.; Mylopoulos, J. (2002) "Towards Requirements-Driven Information Systems Engineering: The Tropos Project." In: The 13th international conference on advanced information systems engineering, Elsevier Science Ltd, v.27, n.6. p. 365-389.

[5] Bresciani, P.; Giorgini, P.; Giunchiglia, F.; Mylopoulos, J.; Perini, A.; TROPOS: An Agent-Oriented Software Development Methodology. In Journal of Autonomous Agents and Multi-Agent Systems. May 2004. Kluwer Academic Publishers.

[6] Chung, L.; Nixon, B.; Yu, E.; and Mylopoulos, J.; Non-Functional Requirements in Software Engineering. Kluwer Academic, 2000.

[7] Kavakli, E. and P. Loucopoulos. Goal-Driven Requirements Engineering: Evaluation of Current Methods. in EMMSAD 2003. 2003. Velden, Austria.

[8] Kavakli, E. Loucopoulos, P.; Goal Modeling in Requirements Engineering: Analysis and Critique of Current Methods. Information Modeling Methods and Methodologies 2005: 102-124

[9] Yu, E., Agent-Oriented Modeling: Software Versus the World In Agent-Oriented Software Engineering AOSE-2001 Workshop Proceedings, Montreal, Canada - May 29th 2001. LNCS 2222.

[10] Estrada, H; Martínez, A; Pastor, O; Mylopoulos, J.; An Experimental Evaluation of the i* Framework in a Model-based Software Generation Environment; E. Dubois, K. Pohl (Eds.); CAISE 2006, LNCS 4001, pp.513-527, 2006. Springer-Verlag, Berlin Heidelberg, ISBN: 3-540-34652-X, 978-3-540-34652-4.

[11] Pastor, Oscar; Estrada, Hugo; Martínez, Alicia; The Strengths and Weaknesses of the i* Framework: an experimental evaluation i*, its Applications, Variations and Extensions. Eric Yu et als. (eds.) MIT Press.

[12] Oliveira, A. Padua A.; Cysneiros, L. M.; "Defining Strategic Dependency Situations in Requirements Elicitation" The IX Workshop on Requirements Engineering; Rio de Janeiro, Brazil - July/2006.

[13] Leite, Julio C. S. P.; Franco, Ana P. M. A Client Strategy for Conceptual Model Acquisition, Proc. of the Intl. Symp. on Requirements Engineering, IEEE Computer Society Press (1993), pp. 243-246.

[14] Goguen, J.A. and Linde, C. - Techniques for Requirements Elicitation, In Proceedings of the First IEEE International Symposium on Requirements Engineering, San Diego, Ca, IEEE Computer Society Press - 1994, pp. 152-164.

[15] Kotonya, Gerald. and Sommerville, Ian. Requirements Engineering – Processes and Techniques – 1998 – John Wiley & Sons Ltd. England.

[16] Yu, E. Agent Orientation as a Modeling Paradigm Wirtschaftsinformatik. 43(2) April 2001. pp. 123-132.

[17] Potts, Colin; Takahashi, Kenji; Antón, Annie I.; "Inquiry-Based Requirements Analysis", IEEE Software, Volume 11, Issue 2 (March 1994), pp. 21 - 32

[18] Cunha, Herbet "Uso de estratégias orientadas a metas para modelagem de requisitos de segurança", Dissertação (Mestrado em Informática) – PUC-Rio, 2007 - 145 f.

[19] Leite, Julio; Werneck, Vera; Oliveira, A. Padua; Capelli, Claudia; Cerqueira, Ana Luiza; Cunha, Herbert; Baixauli, Bruno; "Understanding the Strategic Actor Diagram: An Exercise of Meta Modeling" The X Workshop on Requirements Engineering; Toronto, May/2007.

[20] Leite, Julio C. S. P.; Oliveira, A Padua A.; Client Oriented Requirements Baseline, Proceedings of the Sec. Intl. Symp. on Requirements Engineering, IEEE Computer Society Press, 1995, pp. 108-115.

[21] Zheng You, Using meta-model-driven views to address scalability in i* models, Master of Science thesis, Graduate Department of Computer Science, University of Toronto, 2004, pp. 231.

[22] Hammer, Michael; Champy, James. Reengenharia : revolucionando a empresa em função dos clientes, da concorrência e das grandes mudanças da gerencia / Rio de Janeiro - Campus, 1994. 189p. ISBN 8570018487

[23] Miller, George; The Magical Number Seven, Plus or Minus Two: Some Limits on Our Capacity for Processing Information in The Psychological Review, Harvard University - 1956, vol. 63, pp. 81-97

[24] Ramesh, B.; Jarke, M.; "Towards Reference Models for Requirements Traceability", IEEE TSE V. 21, N 1, 2001.

[25] Mylopoulos, J.; Chung, L.; Yu, E.; `From Object-Oriented to Goal-Oriented Requirements Analysis' Communications of the ACM, 42(1): 31-37, Jan. 1999.

[26] Sharp, A., Mcdermott, P.: Workflow Modeling: Tools for Process Improvement and Application Development. Norwood: Artech House (2000).

[27] Wooldridge, Michael; An Introduction to Multi-Agent Systems, John Wiley and Sons Limited: 2002.