

## A value-based approach for reasoning with goal models

**Carlos Cano-Genoves**

*Universitat Politècnica de València  
Valencia, Spain*

*carcage1@dsic.upv.es*

**Emilio Insfran**

*Universitat Politècnica de València  
Valencia, Spain*

*einsfran@dsic.upv.es*

**Silvia Abrahao**

*Universitat Politècnica de València  
Valencia, Spain*

*sabrahao@dsic.upv.es*

**Marta Fernandez-Diego**

*Universitat Politècnica de València  
Valencia, Spain*

*marferdi@omp.upv.es*

**Fernando González-Ladrón-de-Guevara**

*Universitat Politècnica de València  
Valencia, Spain*

*fgonzal@omp.upv.es*

### Abstract

Goal models are used to represent stakeholders' intentions regarding the system to be developed and the alternative means to achieve those intentions. Goal-oriented analysis techniques have been proposed to help analysts reason when employing goal models. These techniques can be used to identify conflicts among goals, to choose between alternatives or to check the satisfiability of the model. Unfortunately, most of these techniques consider that stakeholders their intentions are equally important. This paper presents a value-based approach that and allows stakeholders to assign a relative importance to the elements in a goal model. It then propagates that importance by means of their relationships (dependencies, contributions and decompositions) in order to determine which elements are more valuable. Fisher's weighted distribution and multi-criteria decision analysis techniques are used to deal with the propagation of the importance among the goals. The contribution is the alignment of information system with stakeholders and organizational goals.

**Keywords:** Goal Model, Value Propagation, Multi-Criteria Decision Analysis

### 1. Introduction

Goal-oriented requirements engineering (GORE) is an important area of Requirements Engineering [9] in which goal models are used to elicit early requirements. These models represent the stakeholders' intentions regarding the system to be developed and the rationale (alternatives) on how to achieve them. Some of the best-known GORE approaches are  $i^*$  [21], the Goal Requirements Language (GRL) [17], and KAOS [7]. Although goal models have been commonly used to elicit requirements, they have also been extended for use in specific domains, as is acknowledged in a recent systematic literature review [13]. For example, in the domain of Social-Technical Systems,  $i^*$  was extended to deal with conflicts of interest in healthcare, while legal aspects were considered when representing regulations in GRL.

Goal analysis techniques are used to help analysts reason about goal models with the

purpose of identifying conflicts among goals, checking the satisfiability of the model, or choosing among alternatives. There are currently many different techniques with which to analyze goal models, and these employ different approaches, such as propagation [2, 16,12,11], simulation [10,20] or planning [6,3].

Unfortunately, like most software development practices [5], these techniques use a value-neutral approach in which business goals, tasks, and resources are equally important. This implies that value<sup>1</sup> is not taken into account downstream of the system development activities and, therefore, the system developed may not meet the organizational goals or stakeholders' expectations. In particular, these techniques do not consider that: i) goals may have different values for the stakeholders; ii) not all the stakeholders (actors) are equally valuable and the importance of their goals may, therefore, vary accordingly, and iii) stakeholders' preferences with respect to the alternatives may have different implications for system development.

Over the past decade, the Value-Based Software Engineering (VBSE) research area emerged putting the concept of value at the forefront of software engineering decisions [5]. In this sense, a field where it could be interesting to apply the principles of ISBV is goal modeling, by prioritizing the different primitives of a model according to the value that it provides to the system stakeholders.

The purpose of this paper is, therefore, to propose a value-based analysis approach with which to reason when using goal models. This approach makes use of value propositions in order to prioritize the different modeling primitives of the goal model (actors, intentional elements, and relationships) by assigning a relative importance (value proposition), which is then propagated by means of the dependency, contribution, and decomposition relationships in the goal model. The value proposition is primarily used as a generic term that encompasses win conditions or any aspect of interest (tangible or intangible, economic or social, monetary or utilitarian, and aesthetics or ethics) from a given stakeholder's or organization's point of view.

Our approach, therefore, makes it possible to align information systems with stakeholders' and organizational goals. The proposed approach specifically provides a means to reason about the relative importance of goals that can be inherited by the system design and development activities. This potentially improves the stakeholders' perceived value of the system by increasing the likelihood that those stakeholders' most important goals will be dealt with first.

The remainder of this paper is organized as follows. Section 2 discusses existing goal analysis techniques. Section 3 presents the proposed value-based analysis technique used to reason with goal models and the propagation algorithms that supports it, while Section 4 presents an illustrative example that demonstrates the feasibility of our approach. Finally, Section 5 concludes this paper and summarizes directions for further work.

## 2. Related work

Goal analysis techniques can be classified on the basis of the approach used to reason about goal models. These techniques can be classified into several categories, such as systematic propagation, simulation, planning or multi-criteria decision analysis (MCDA). However, in this paper, we focus on discussing the systematic propagation and MCDA approaches, as they are those which are most closely related to our proposal.

### 2.1. Systematic propagation

One of the approaches most frequently used in goal analysis is the systematic propagation of goal satisfaction, which can be used to answer questions such as "Will a particular design alternative work in the domain?" or "What are the consequences of its

<sup>1</sup> Value is traditionally seen as a profit generation activity. However, and as acknowledged by Khurum et al. [18], it is a much more complex concept that greatly relies on stakeholders' or an organization's point of view.

implementation?” [15]. This approach is based on the assignment of goal satisfaction and its propagation by means of relationships. Furthermore, depending on the direction of the propagation, it will have a different use and will answer different questions.

On the one hand, if the propagation is made from “leaf to root” (forward propagation), the approach will answer the question “What if?” in order to discover what will occur if that leaf (alternative) is chosen, i.e., it shows the impact that one alternative will have when compared to the other intentional elements in the goal model.

On the other hand, if the propagation is made from “root to leaf” (backward propagation) the approach will answer the question “Is it possible?” in order to discover whether it is possible to satisfy the initialized goal [2]. Backward propagation is used to find the set of goals at the minimum cost that, if achieved, can guarantee the achievement of the desired goals.

The techniques that use this approach [2,16,12,11] allow us to know how an intentional element or a group of them affect the model. There are two main drawbacks with this approach: i) the propagation should be done with every possible combination of intentional elements, which can cause problems with large models, and ii) most of these techniques do not consider the stakeholders’ preferences and they require the analyst’s collaboration in order to decide which is the best combination of intentional elements.

## 2.2. Multi-Criteria Decision Analysis (MCDA)

Several proposals with which to analyze goal models through the use of MCDA have appeared in recent years [22,4]. MCDA has been widely employed in many fields to make decisions and has been fully discussed and validated. An MCDA approach in goal analysis consists of evaluating the degree of satisfaction that each alternative provides for any selected criterion. This type of analysis does not usually consider relationships between intentional element nor between criteria.

Unlike the systematic propagation approaches, these techniques consider the stakeholders’ preferences. However, none of them considers that different stakeholders may have a different degree of importance. Some of the limitations of these techniques are: some of them [22] have scalability problems owing to the MCDA technique used, since they have to compare all the intentional elements and relationships in pairs, while others [4] do not consider the existing relationships between the intentional elements and do not, therefore, consider how an intentional element can affect the model.

## 3. The GATHA approach

The Goal-oriented Analysis THrough vAlue (GATHA) approach aims to help analysts and stakeholders align information systems with stakeholders’ and organizational goals by providing a value-based approach in which actors, intentional elements and relationships are prioritized and then propagated by means of the model. Although the approach can be applied to goal models by following the  $i^*$  and its variants (e.g., GRL or Tropos), in this paper, we use the GRL notation to illustrate how the approach can be used. The approach consists of two main activities, prioritization and propagation, as shown in Fig. 1. In the following, we first introduce the main concepts of the GRL language and then describe these activities in detail.

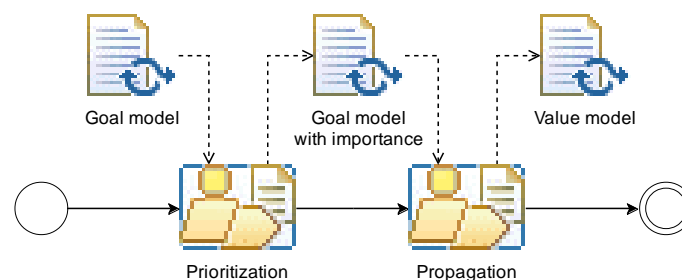


Fig. 1. The GATHA technique

### 3.1. The GRL language

The Goal-oriented Requirements Language (GRL) is part of the User Requirements Notation (URN) standard [17]. GRL aims to capture business or system goals, (sub)goals and tasks that help achieve high-level goals. There are three categories of concepts in GRL: actors, intentional elements, and relationships.

*Actors* represent entities (stakeholders or systems) in the domain of interest, which have intentions and may perform actions to achieve their objectives.

*Intentional elements* describe the intention and capabilities of an actor. There are three types of intentional elements: i) *goal*, which represents a condition or state of affairs about the system to be developed that an actor would like to achieve; ii) *softgoal*, which is a more abstract condition than a goal and there is no clear measure to verify its satisfaction. Usually, softgoals are often used to describe quality (i.e., *happy customer*) or non-functional requirements; iii) *task*, which captures a solution to achieve goals or softgoals by means of actions to be performed.

*Relationships* are used to connect intentional elements. There are three types of relationships: i) *contribution relationship*, which represents the impact of one intentional element on another element. This impact may be either positive (+) or negative (-) within the same actor or between different actors; ii) *decomposition relationship*, which allows an intentional element to be decomposed into sub-elements (using AND, OR, or XOR) within the same actor; iii) *dependency relationship*, which models the relationship between intentional elements of different actors. This means that the satisfaction or realization of one depends on the satisfaction or realization of the other.

### 3.2. Prioritization activity

The prioritization activity consists of determining the degree of importance for some stakeholders of the different primitives in a goal model in one or more value dimensions (e.g., personal preference, business value, cost reduction, etc.).

The input for this activity is a goal model without cycles. A goal model is acyclic or does not have cycles when there is no intentional element that can be reached by means of relationships. The reason why the model must be without cycles is because the propagation used in our approach would not end if there were.

In this activity, actors, intentional elements, and relationships are prioritized by means of the assignment of a relative importance, in which each of them can have one of the following degrees of importance: Irrelevant (0), Low (25), Medium (50), High (75), and Indispensable (100). It is also possible to assign a degree or level of importance between the values of 0 and 100. This activity is composed of three tasks, each of which is responsible for prioritizing the following primitives of a goal model: actors, intentional elements and relationships. Our proposal is concerned with the propagation of the relative importance in order to calculate the value that each intentional element has without being bound to a particular method to prioritize the acquisition of relative importance.

In the *prioritizing actors* task, a relative importance (value proposition) is assigned to each actor (stakeholder) in the goal model, since each stakeholder may have a different level of importance. The importance of the actors should be assigned by analysts following their own criteria (e.g., economic, strategic, performance, social).

In the *prioritizing intentional elements* task each intentional element is assigned an importance (value proposition) through negotiation with the stakeholder to which the intentional element belongs. Since the assignment of importance can have different meanings depending on the element type and its belonging to a decomposition, we propose the following rules to assist stakeholders when performing this task:

- If the intentional element type is a softgoal, or is not decomposing another element, the importance must answer the question: *How important am I for the actor to which I belong?*
- If the intentional element is decomposing another element, and is not a softgoal, the importance must answer the question: *How important am I for the element*

*that I decompose?*

In the *prioritizing relationships* task, an importance (value proposition) is assigned to each relationship between intentional elements. The meaning of the importance varies depending on the type of relationship:

- If the relationship is a dependency, the importance represents the degree of dependence from one element to another.
- If the relationship is a contribution, the importance represents the degree of contribution.
- Decomposition relationships are considered during the *prioritizing intentional elements* task.

Contribution relationships can have negative (-) or positive (+) importance because they can contribute positively or negatively to another element. In addition, changing the importance of contribution links can, according to the scores proposed in the Z.151 standard [17], have an impact on the type of contribution, as indicated in Table 1.

**Table 1.** Contribution made by type to importance

Contribution type	Importance
Make	100
SomePositive	75
Help	25
Unknown	0
Hurt	-25
SomeNegative	-75
Break	-100

The output of the prioritizing activity is a goal model with importance, in which actors, intentional elements, and relationships have been prioritized according to the value that they have for the different stakeholders.

### 3.3. Propagation activity

This activity consists of propagating the importance that each actor and intentional element has by means of the relationships in the model in order to calculate the value of each intentional element. The input for this activity is an acyclic goal model in which the importance of each actor, relationship, and intentional element has already been made explicit.

In the following, we explain how the importance should be propagated by means of actors and relationships and the order in which that importance is propagated through the use of relationships.

#### *Propagation of importance through the use of relationships*

Because not all the *stakeholders (actors)* are equally important, the importance assigned to them affects the importance of their components. This is dealt with by using Eq. 1, in which the importance of the actor (A), which is given a value of between 1 and 100 (the maximum importance that can be assigned to it) is multiplied by the importance of the intentional element (IE). The result is the value that the intentional element has for the actor to which it belongs without considering the relationships.

$$value(IE) = \frac{A}{100} * IE \quad (1)$$

Thanks to the value of the intentional elements, it is possible to consider the importance of the actor to whom they belong. As an example of propagation, if an intentional element with an importance of 100 belongs to an actor with an importance of 50, the value of the intentional element after propagating the actor will be 50.

The propagation of importance (value proposition) in a decomposition relationship is carried out in both directions, top-down and bottom-up. The former distributes the value of the decomposed intentional element (parent) between the elements that decompose it (children), while the latter modifies the value of the parent intentional element based on the “behaviour” (relationships) of its children.

The decomposition is propagated in a top-down direction by applying Eq. 2, which is based on Fisher’s weighted distribution [8]. When using this equation, the value that one child gets depends on its importance to its parent (IE), the importance of its parent (pIE) and the importance of all the children of the decomposition (sIE). One special feature of using this propagation is that the sum of the value of all the children is equal to the value that the parent distributes.

$$value(IE) = \frac{IE}{\sum_{d=1}^n sIE_n} * pIE \quad (2)$$

The more children a decomposition has, the lower the value they will have, while the more importance an intentional element has, the higher the value it will gain. This is shown in Fig. 2, in which the number between parenthesis is the importance (value proposition) and that between brackets is the value after propagation.

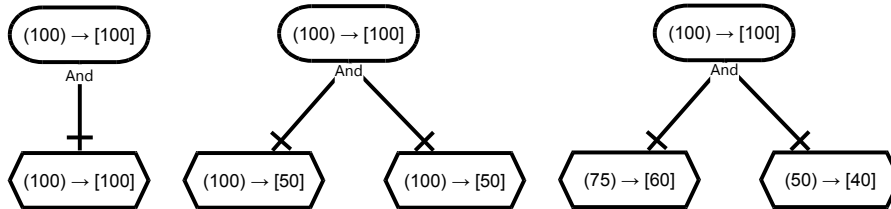


Fig. 2. Examples of top-down propagation in decomposition relationships

This propagation procedure makes it possible to consider the preferences that each stakeholder has as regards the alternatives. Furthermore, it also makes it possible to consider that there may be intentional elements in a decomposition of different levels of abstraction and, therefore, different importance (value proposition).

The bottom-up decomposition is propagated using the feature of the top-bottom propagation in which the sum of the value of all the children is equal to the value that the parent distributes. This signifies that when one child gains or loses value, the parent will gain or lose the same value. For example, if one child gains or loses value thanks to a contribution his parent will gain or lose the same value thanks to that contribution.

Softgoals are not considered during the propagation of decomposition relationships because they are used to guide (or restrict) the selection among alternatives, rather than representing a particular course of action.

Fig. 3 shows the algorithm used to propagate the value in a decomposition from parent to children (top-down). In order to use this propagation, decomposition relationships include an attribute to indicate whether the propagation has been made from *parent2children*, from *children2parent* or both. The algorithm employed to propagate a decomposition from child to parent (bottom-up) has been included in the algorithm for propagating contribution and dependency relationships because the child to parent propagation is made when the child modifies its value.

**Algorithm:** Decomposition propagation from parent to children  
**Input:** ie:IntentionalElement

```

sum:Integer = 0;
for each dec:Decomposition in ie.linksDest
  if (dec.src.type == SOFTGOAL)
    dec.propagated = true //Mark as propagated
  else
    sum += dec.src.importance;//Calculate importance of children

for each dec:Decomposition in ie.linksDest //Distribute among children
  if (dec.src.type != SOFTGOAL && dec.status == NOTHING){
    dec.src.value = (dec.src.importance / sum ) * dec.dest.value
    dec.status = PARENT2CHILD //Mark as propagated
  }

```

**Fig. 3.** Example: Decomposition propagation algorithm from parent to children

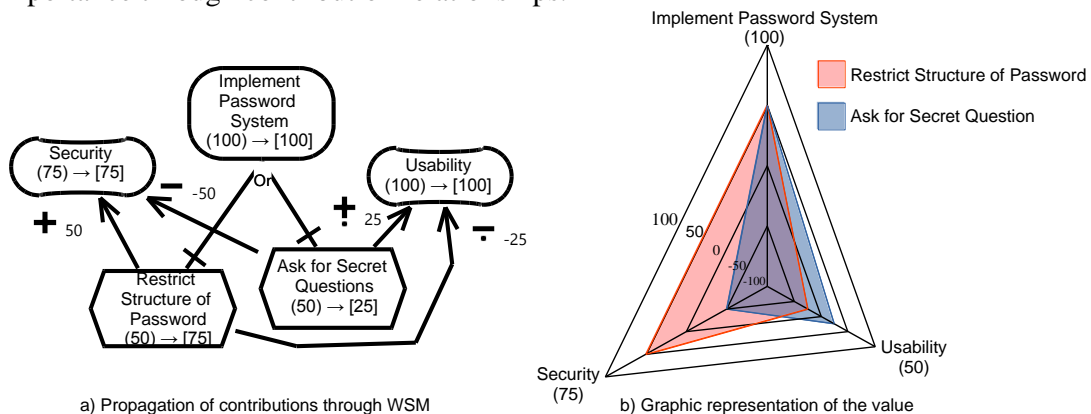
The propagation from parent to children (top-down) must be done before that the child to parent (bottom up) to prevent a child affecting other children of the decomposition, for this we have included exceptions in the algorithm of propagation of dependencies and contributions.

The propagation of importance (value proposition) through the use of *contribution relationships* is carried out by means of multi-criteria decision analysis techniques [19]. In this paper, we specifically use the Weighted Sum Model (WSM), but other techniques such as the Weighted Product Model (WPM) or the Analytic Hierarchy Process (AHP) could have been used to propagate the importance of contribution relationships.

The WSM technique is used to compare alternatives or options (A) by assigning a weight (W) to each (n) criteria and a weight to each performance (P) of the option for each criterion. In our approach, we divide weight by 100 because this is the maximum weight that can be assigned, as indicated in Eq. 3.

$$A_i = \sum_{j=1}^n \frac{W_j}{100} P_{ij} \quad (3)$$

In order to use it in a goal model, we have considered the source of a contribution as the “option”, the destination as the “criterion” and its importance as its “weight”, and the type of contribution (how much the source contributes to the destination) as the “performance of that option for that criteria”. Fig. 4 a) shows an example of the WSM application in a goal model based on [14], in which the alternative *Restrict Structure of Password* has more value than *Ask for Secret Question*. The reason for this is graphically represented in Fig. 4 b). Fig. 5 shows an example of the algorithm used to propagate the importance through contribution relationships.



**Fig. 4.** Examples of propagation in contribution relationships

```

Algorithm: Contribution propagation
Input: ie:IntentionalElement

for each cont:Contribution in ie.linksDest {
  if (cont.propagated == true)
    continue

  //EXCEPTION: Do not propagate to an element that has not get value from its parent
  if (cont.src.has(dec:Decomposition | dec.status == NOTHING))
    continue

  cont.src.value += (cont.quantitativeContribution / 100) * ie.value
  cont.propagated = true

  //Propagate decomposition from child to parent
  for each dec:Decomposition in cont.src.linksSrc
    if (dec.src.type != SOFTGOAL) //Do not propagate softgoals
      dec.dest.value += (cont.quaitativeContribution/ 100) * ie.value
}

```

**Fig. 5.** Example: Contribution propagation algorithm

We have used the accumulative value equation from MAGERIT (Methodology for Information Systems Risk Analysis and Management) [1] as a basis for the propagation of importance (value proposition) through the use of *dependency relationships*. Our reason for using this equation is because dependency relationships have been employed extensively in risk analysis and researched in depth. The value that an intentional element that is dependent on another has is calculated by means of Eq. 4., in which the dependent Intentional Element (IE) attains the value of the dependent Intentional Element (dIE) by considering the degree of dependency (degree) divided by 100, which is the maximum degree of dependency. For example, if one intentional element with an importance of 100 is depended on by another one with an importance of 50 with a degree of dependency of 50, the first one will attain 25.

$$value(IE) = \sum_i \left\{ value(dIE_i) * \frac{degree(dIE_i \Rightarrow IE)}{100} \right\} \quad (4)$$

The original equation of accumulative value considers the transitive property, in which if one intentional element is depended on by another which is depended on by yet another, the element will attain value from both dependencies. Our proposed equation for the propagation of importance in dependency relationships does not consider the transitive property directly, but indirectly, as can be seen below. Fig. 6 shows an example of the algorithm used to propagate the importance through dependency relationships.

```

Algorithm: Dependency propagation
Input: ie:IntentionalElement

for each dep:Dependency in ie.linksSrc {
  if (dep.propagated == true)
    continue

  //EXCEPTION: Do not propagate to an element that has not get value from its parent
  if (dep.dest.has(dec:Decomposition | dec.status == NOTHING))
    continue

  dep.dest.value += ( dep.degreeOfDependency/ 100 ) * ie.value
  dep.propagated = true

  //Propagate decomposition from child to parent
  for each dec:Decomposition in dep.dest.linksSrc
    if (dec.src.type != SOFTGOAL) //Do not propagate softgoals
      dec.dest.value += ( dep.degreeOfDependency/ 100 ) * ie.value
}

```

**Fig. 6.** Example: Dependency propagation algorithm



### *Order of propagation*

The propagation of importance by means of the relationships in the model must be carried out in a specific order so as to avoid indeterminism (more than one result for the same intentional element), and to include transitivity (one intentional element can affect another indirectly). We have, therefore, developed an algorithm with which to indicate the order of propagation when considering both.

Fig. 7 shows the algorithm used to execute the propagation in an orderly manner, such that indeterminism is avoided but transitivity is included. First, the actors' importance is propagated, after which the relationships between intentional elements are propagated in an orderly manner so that those intentional elements that cannot gain more value are propagated first and those intentional elements that can gain value are not propagated until they attain all the possible value.

**Algorithm:** Ordered propagation algorithm

**Input:** GRLmodel:GRLspec

```

elements:List = ∅ //intentional elements to be propagated
propagateActors(GRLmodel)
for each actor:Actor in GRLmodel.actors
for each ie:IntentionalElement in actor.elems
elements.add(ie)
while (elementsReady.size() > 0) {
    ie = elements.get()
    elements.remove(ie)
    canPropagate:Boolean = true

    //Check if the element can get value from dependencies or contributions
    if (ie.linksDest.exists(dep:Dependency | dep.propagated == false) ||
        ie.linksSrc.exists(cnt:Contribution | cnt.propagated == false))
        canPropagate = false

    //Check if the element can get value from his parent (decomposition)
    if (ie.linksSrc.exists(dec:Decomposition | dec.status == NOTHING))
        canPropagate = false

    //Do not propagate if the element can change its value
    if (canPropagate == false) {
        elements.add(ie)
        continue
    }

    //Propagate decomposition from parent to children
    propagateDecomposition(ie)

    //If the element is a parent and can get value from its children do not propagate
    if (ie.linksDest.exists(dec:Decomposition |
        dec.status == PARENT2CHILD)) {
        elements.add(ie)
        continue
    }

    propagateDependencies(ie)
    propagateContributions(ie)

    //Check if the element has finished propagating
    if (ie.linksSrc.exists(dep:Dependency | dep.propagated == false) ||
        ie.linksDest.exists(cnt:Contribution | cnt.propagated == false) {
        elements.add(ie)
        continue
    }

    //Confirm de propagation of the decomposition from child 2 parent
    for each dec:Decomposition in ie.linksSrc {
        dec.status = CHILD2PARENT
        dec.propagated = true
    }
}

```

**Fig. 7.** Example: Order of the propagation algorithm

The output of the propagation activity is a goal model in which intentional elements have importance and value, and we have called this a value model because it represents the value that each intentional element has.

### 4. Illustrative example

The use of the proposed approach is illustrated in a scenario introduced by Giorgini et al. [11], where the strategic objectives of a US car manufacturer, such as GM, are represented by means of a goal model. In the example, the objectives of the manufacturers are to sell vehicles with the maximum benefits. However, in this paper, we present an extension of the original goal model in which the goals and preferences of the related stakeholders are also represented. For example, the customer wants to buy a high-quality car.

Following the proposed approach, the first step is the prioritization, during which the analyst has to prioritize the stakeholders, after which the stakeholders have to prioritize their intentional elements and their relationships by answering the questions mentioned in Section 3.1. For example, if the customer’s aim is to buy a car, the goal “buy a car” is indispensable (100), the security of the car has a high importance (75) for the customer and the quality is also important but not as much as safety (Medium importance [50]). Fig. 8 shows the goal model with prioritization in which the number between parenthesis is the relative importance (value proposition) assigned.

The second step of the proposed approach is the propagation, during which the assigned importance is propagated by considering the stakeholders’ (actors) importance and the relationships between intentional elements following the order of propagation. For example, for the customer’s goal of buying a car, the importance of 100 before the propagation is reduced to 75 after the propagation since the importance of the stakeholder (customer) makes it lose importance. Fig. 8 shows the goal model after the propagation, in which the number between parenthesis is the relative importance (value proposition) assigned and the number between brackets is the value that the intentional element has. The numbers between parenthesis and brackets indicate the order of propagation, if there are two relationships with the same number it is because it can be done in any order.

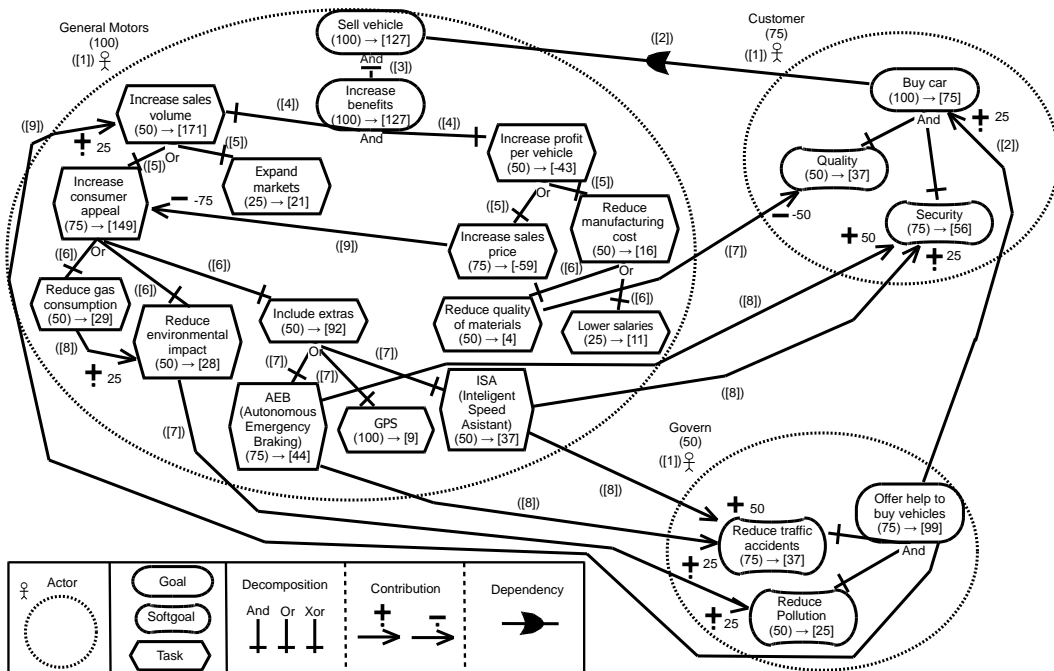


Fig. 8. Goal model with assigned importance

This goal model with value can be used to reason about the best strategy by which to sell vehicles by the manufacturer with the maximum benefits. For example, in the case

that the manufacturer wants to *increase the profit per vehicle*, the best option is to *reduce the manufacturing cost*. The reason for this is that *increasing the sales price* loses value owing to the negative effect on the *increase in consumer appeal*. Customers do not like expensive cars, and the way in which to achieve the *reduction in manufacturing costs* is by *lowering salaries*, because *reduce the quality of materials* has a negative impact on the *quality* of the car, which is important for customers.

## 5. Conclusions

In this paper we have presented a value-based approach for use when reasoning about goal models. The approach makes it possible to establish the relative importance (value proposition) of the different primitives in a goal model according to the stakeholder's point of view, taking into account the relationships among these elements. The relative importance is then propagated by means of the model in order to obtain the corresponding value.

This approach can help analysts make decisions by considering the value that each intentional element (or alternative) has, which is interesting because most of the techniques used to reason about goal models focus on goal satisfaction and do not consider stakeholders' preferences. The main contribution of our approach is a means to reason about the relative importance of goals that can be inherited by the system design and development activities. In addition, it facilitates the alignment of information systems with stakeholders and organisational goals. For example, this approach can be used to select the software increments that will be delivered first in a continuous delivery development process. We are currently defining an Eclipse-based environment to automate the approach.

There are several limitations that deserve attention. The illustrative example may not reflect the complexity of real-world cases or how our approach could be beneficial in these cases. However, we consider this as a preliminary approach to the problem of how to deal with value in goal models. Another limitation is that our approach assumes that a goal model must be without cycles. In future work, we plan to study how to deal with cycles in goal models. Moreover, the approach does not explicitly manage the evolution of goal models by taking into account changes in the different stakeholders' preferences. We plan to study how these models can be continuously updated to support decision making and to keep the corresponding information system updated.

As future work, we also plan to analyze the interaction between intentional elements in greater depth when they are used as possible alternatives. This is because we believe that the value that an intentional element has could change depending on how it relates to other alternatives. We also believe that it would be interesting to consider both value and satisfaction, because there may be intentional elements (alternatives) that have a high amount of value but do not satisfy others, or intentional elements that have low value but satisfy everything. Finally, we plan to perform case studies or controlled experiments in order to evaluate the effectiveness of our approach in practice.

## Acknowledgements

This work was supported by the Spanish Ministry of Science, Innovation and Universities [Adapt@Cloud, grant number TIN2017-84550-R] and the "Programa de Ayudas de Investigación y Desarrollo" (PAID-01-17) from the UPV.

## References

1. Amutio, M.A., Candau, J., Mañas, J.: Magerit-version 3, methodology for information systems risk analysis and management, book I-the method. Minist. Adm. públicas. (2014)
2. Amyot, D., Ghanavati, S., Horkoff, J., Mussbacher, G., Peyton, L., Yu, E.: Evaluating goal models within the goal-oriented requirement language. *Int. J. Intell. Syst.* 25 (8),

- 841–877 (2010)
3. Asnar, Y., Bryl, V., Giorgini, P.: Using risk analysis to evaluate design alternatives. In: *International Workshop on Agent-Oriented Software Engineering*. pp. 140–155. Springer (2006)
  4. Baslyman, M., Amyot, D.: A Distance-Based GRL Approach to Goal Model Refinement and Alternative Selection. In: *2017 IEEE 25th International Requirements Engineering Conference Workshops (REW)*. pp. 16–20. IEEE (2017)
  5. Boehm, B.W.: Value-Based Software Engineering: Overview and Agenda. In: *Value-Based Software Engineering*. pp. 3–14. Springer Berlin Heidelberg, Berlin, Heidelberg (2006)
  6. Bryl, V., Giorgini, P., Mylopoulos, J.: Supporting requirements analysis in tropos: a planning-based approach. In: *Pacific Rim International Conference on Multi-Agents*. pp. 243–254. Springer (2007)
  7. Dardenne, A., Van Lamsweerde, A., Fickas, S.: Goal-directed requirements acquisition. *Sci. Comput. Program.* 20 (1–2), 3–50 (1993)
  8. Fisher, R.A.: The effect of methods of ascertainment upon the estimation of frequencies. *Ann. Eugen.* 6 (1), 13–25 (1934)
  9. Franch, X., Maté, A., Trujillo, J.C., Cares, C.: On the joint use of i\* with other modelling frameworks: A vision paper. In: *2011 IEEE 19th International Requirements Engineering Conference*. pp. 133–142. IEEE (2011)
  10. Gans, G., Schmitz, D., Arzdorf, T., Jarke, M., Lakemeyer, G.: SNet reloaded: Roles, monitoring and agent evolution. In: *Agent-Oriented Information Systems II*. pp. 68–84. Springer (2005)
  11. Giorgini, P., Mylopoulos, J., Nicchiarelli, E., Sebastiani, R.: Formal reasoning techniques for goal models. In: *Journal on Data Semantics I*. pp. 1–20. Springer (2003)
  12. Giorgini, P., Mylopoulos, J., Sebastiani, R.: Goal-oriented requirements analysis and reasoning in the tropos methodology. *Eng. Appl. Artif. Intell.* 18 (2), 159–171 (2005)
  13. Goncalves, E., Castro, J., Araújo, J., Heineck, T.: A systematic literature review of istar extensions. *J. Syst. Softw.* 137 1–33 (2018)
  14. Horkoff, J., Eric, S.K.: Qualitative, Interactive, Backward Analysis of i\* Models. In: *iStar*. pp. 43–46. Citeseer (2008)
  15. Horkoff, J., Yu, E.: Comparison and evaluation of goal-oriented satisfaction analysis techniques. *Requir. Eng.* 18 (3), 199–222 (2013)
  16. Horkoff, J., Yu, E.: Evaluating goal achievement in enterprise modeling—an interactive procedure and experiences. In: *IFIP Working Conference on The Practice of Enterprise Modeling*. pp. 145–160. Springer (2009)
  17. ITU-T – International Telecommunications Union: Recommendation Z.151 (10/18) User Requirements Notation (URN) – Language definition, <https://www.itu.int/rec/T-REC-Z.151-201810-I/en>, Accessed: March 04, 2019, (2018)
  18. Khurum, M., Gorschek, T., Wilson, M.: The software value map—an exhaustive collection of value aspects for the development of software intensive products. *J. Softw. Evol. Process.* 25 (7), 711–741 (2013)
  19. Triantaphyllou, E.: Multi-criteria decision making methods. In: *Multi-criteria decision making methods: A comparative study*. pp. 5–21. Springer (2000)
  20. Wang, X., Lespérance, Y.: Agent-oriented requirements engineering using ConGolog and i\*. In: *Agent-Oriented Information Systems Workshop (AOIS-2001)*. Montreal, Canada. pp. 59–78. (2001)
  21. Yu, E.S.K.: Towards modelling and reasoning support for early-phase requirements engineering. In: *Proceedings of ISRE '97: 3rd IEEE International Symposium on Requirements Engineering*. pp. 226–235. (1997)
  22. Zhao, T., Zhao, H., Zhang, W., Jin, Z.: Goal model driven alternative selection: a quantitative approach. In: *2015 IEEE International Model-Driven Requirements Engineering Workshop (MoDRE)*. pp. 1–10. IEEE (2015)