Specifying Value in GRL for Guiding BPMN Activities Prioritization

Emilio Insfrán *einsfran@dsic.upv.es*
Universitat Politècnica de València
Valencia, Spain

Silvia Abrahão *sabrahao@dsic.upv.es*
Universitat Politècnica de València
Valencia, Spain

Raphael Pereira de Oliveira *raphael.oliveira@ifs.edu.br*
Federal Institute of Sergipe
Sergipe, Brazil

Fernando González-Ladrón-de-Guevara *fgonzal@omp.upv.es*
Universitat Politècnica de València
Valencia, Spain

Marta Fernández-Diego *marferdi@omp.upv.es*
Universitat Politècnica de València
Valencia, Spain

Carlos Cano-Genoves *carcage1@inf.upv.es*
Universitat Politècnica de València
Valencia, Spain

Abstract

In a value-based requirements engineering approach, the stakeholders' value propositions must be considered ahead to drive the software development activities. Early requirements approaches like the Goal-Oriented Requirements Language (GRL) focus on modeling goals to satisfy the stakeholders' needs, but do not provide a proper mechanisms to specify value according to stakeholders' value propositions. Moreover, in software development, after specifying value propositions, there is a need to align goal elements with business process elements in order to prioritize which business process activity will be developed next. Thus, we propose a new approach (value@GRL) to improve GRL with the stakeholders' value propositions and, consequently, the prioritization of Business Process Model and Notation (BPMN) activities, according to the mapping between GRL and BMPN elements. Value@GRL provides guidelines for value specification in GRL models in regards of the stakeholders' needs and the prioritization mechanisms to define the order of BPMN activities in the context of an incremental software development.

Keywords: Value-based development, GRL, prioritization, early requirements, BPMN.

1. Introduction

According to Boehm [6], much of current software engineering practice and research is done in a value-neutral setting. In the context of requirements engineering, business processes, requirements, use cases, and so on, are considered equally important during the elicitation and specification stages and there are no clear means to satisfy customers needs through early and continuous delivery of valuable software. Value has been traditionally seen as a profit generation activity, however, it is a much more complex concept as described by Khurum et al. [13] and highly relies on stakeholder's or organization's point of view.

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) to a given stakeholder or organization. Such value propositions are difficult to be tackle in early stages of the software development. However, early requirements approaches, such as goal-oriented modeling, have been recognized to be useful to understand the organizational context of a software system, the goals, and the dependencies among stakeholders [3]. In addition, goal models may be enhanced when are exploited during other phases of the development process and are used as part of the entire system life cycle.

A well-known early requirements approach is the Goal-Oriented Requirements Language (GRL) [11]. GRL supports the specification of goals, tasks, contributions, and dependencies among actors to satisfy the stakeholders' needs. GRL is recognized as an international standard for goal-oriented modeling and it is part of the User Requirements Notation (URN). Although GRL has some mechanisms to propagate evaluations of some elements [3], it does not provide ways to specify stakeholders' value propositions. Moreover, since GRL is an early requirements approach, there is also a need to align the information of goal-oriented models with lower level models of the software development, such as business process models. Horkoff et al. [10] performed a systematic review on goal modeling languages to better understand how they can be integrated into downstream system development artifacts. They found that although much work has been done in this area, it is still fragmented, following separate strands of goal-orientation, and is often in early stages of maturity.

In lower phases of the software development, after the specification of the goal model, there is a need to specify the businesses processes, which is an activity well accepted by the industry [13]. The Business Process Model and Notation (BMPN) [14] is an example of technology used in this situation. BMPN has some elements such as activities, events, control flow, and others to capture the process activities flow. Following the specification of a BPMN model, according to the incremental software development, there is a need to prioritize the activities that will be aligned with another lower level phase of the software development, such as the requirements.

Thus, in order to cope with the value proposition specification in goal modeling and the prioritization of business process activities, this paper introduces an approach (value@GRL) to support: (1) the modeling of value propositions in GRL; and (2) the alignment of GRL elements and BPMN elements to prioritize BPMN activities in the context of incremental software development. Through guidelines, value@GRL allows to represent value propositions for stakeholders or organizations, taking into account the information system (IS) to be developed. In the context of incremental software development, this value model can then be used to prioritize business processes in order to provide decision criteria on which activities of the business processes will be included in a given increment, integrating, in this way, the value model into downstream artifacts of the system development.

The structure of this paper is as follows. Section 2 introduces our approach including the notational elements, the related process for creating a value model, and the mechanisms for propagating the value among the model elements. Section 3 presents a set of rules for deriving and prioritizing the activities of a BPMN model using the value model. Section 4 discusses related work. Finally, Section 5 outlines our conclusions and further work.

2. Specifying Value in GRL

Value@GRL is a specialization of GRL that supports the value modeling regarding the information system to be developed. The justification to make GRL more specific is that we do not intend to define a new language, instead, we want to take advantage of the existing knowledge on goal modeling to provide a systematic process towards a more effective and practical way to specify the stakeholders' value propositions when developing information systems. GRL is a rich goal-oriented language that can be used for different purposes such as social modeling, decision-making, and rationale documentation. However, with the purpose of specifying value propositions, a GRL subset has just been selected. Thus, as our focus is on modeling value, we removed some elements from the original GRL, as explained next.

2.1. Value@GRL Process Overview

The process for dealing with value propositions in value@GRL involves four main activities (goal modeling, goal model prioritization, high-level business process modeling, and business process prioritization), as shown in Figure 1.

The first activity (goal modeling) is responsible for creating a goal model according to the guidelines of the method. Then, it is possible to follow two alternative paths: run the value@GRL algorithms and propagate the value among the intentional elements (goal model value propagation); and generate the high-level business process model (BPMN) from the value@GRL goal model (high-level business process modeling). The last activity (business process prioritization) is concerned with the prioritization of the activities from the business process. The notation for creating the goal model using value@GRL is shown below.

2.2. Value@GRL Notation

Since value@GRL is a specialized subset and specialization of the elements of GRL, its notation has been reduced. There are three categories of concepts in value@GRL: actors, intentional elements, and links.

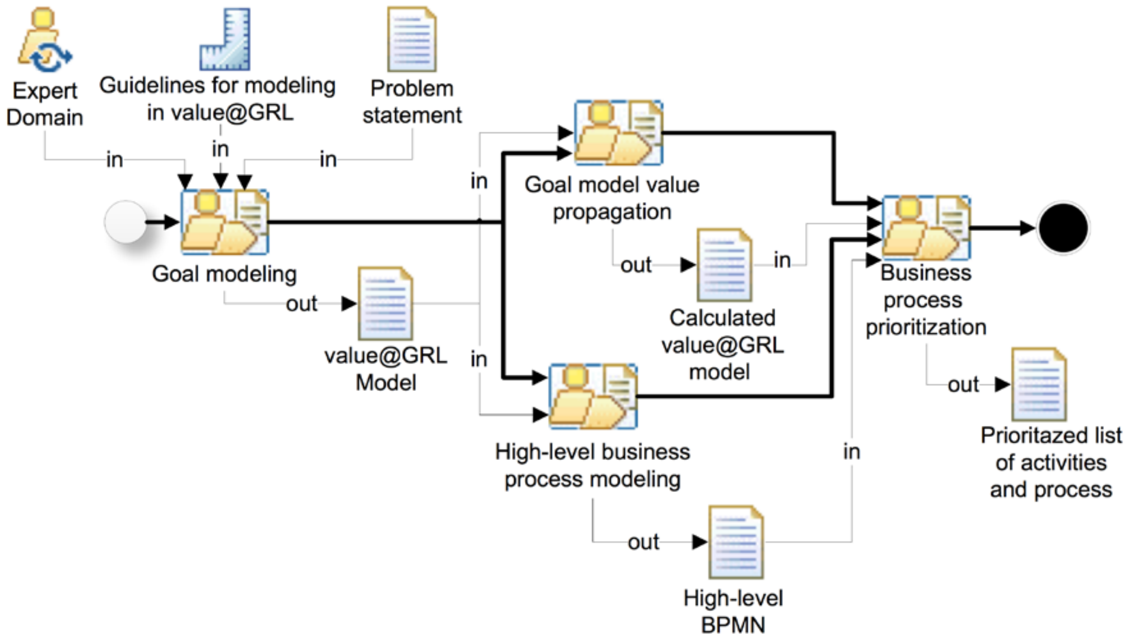

Fig. 1. Process definition for value@GRL.

Actors represent active entities (stakeholders or systems) in the domain of interest, which have intentions and may perform actions to achieve their objectives. A difference from GRL is that value@GRL defines three types of actors:

- **Main actor**. It represents the stakeholder that will drive the specification of the value model. It is the main stakeholder for which the system is to be developed. Usually, there is only one main actor in each value model. However, two or more main actors can also be identified according to the domain. This actor is labeled with the tag «main».
- **External actor**. It represents collaborators or affected stakeholders who have values that the system actor may take in consideration to satisfy its own values. They are labeled with the tag «external».
- **System actor**. It represents the system to be developed including the set of values and operations needed to satisfy the objectives of the involved actors. It is labeled with the tag «system».

The different types of actors helps in identifying the system to be developed («system») for the main actor («main») and, how external actors («external») will assist the system to achieve its goals.

Intentional elements describe the intention and capabilities of an actor. They are always represented inside the boundary of a given actor. Value@GRL uses three types of intentional elements:

- **Goal**. It is a condition or state of affairs about the system to be developed that an actor would like to achieve.
- **Soft-goal**. It is a condition or state of affairs regarding the system to be developed that an actor would like to achieve. A soft-goal is, however, more abstract than a goal and there is no clear measure or satisfaction condition to verify. Usually, soft-goals are later related to non-functional requirements of the system to be developed.
- **Task**. It captures a solution to achieve goals or soft-goals by means of actions to be performed.

Links are used to connect intentional elements. Value@GRL has three types of links:

- **Contribution link**. It models the desired impact of one intentional element on another element. This impact may be either positive (+) or negative (-) within the same actor or between an actor and the system actor.
- **Decomposition link**. It allows an intentional element to be decomposed into sub-elements (using AND, OR, or XOR) within the same actor.
- **Dependency link**. It models the relationship between intentional elements of different actors. This means that the satisfaction or realization of one intentional element depends on the satisfaction or realization of the other. Only dependencies between the system actor and the other actors need to be modeled.

It is important to highlight that in a value@GRL model, all actors (main actor and external actors) need to interact with the system actor through dependencies and/or contribution links, since the main purpose of the value@GRL model is to reason about the different value propositions of different actors who interact with the system to be developed.

Figure 2 shows an excerpt of a value@GRL model for a mobile application, called Green Route, which was initially modeled with i* [9] (the numbers found in this model inside intentional elements and also along contribution links will be explained next in sections 2.3 and 2.4). Green Route application proposes the optimal route for a user, avoiding routes with high levels of pollution, flood, pollen, etc., allowing, for instance, obtaining the preferred routes for people with respiratory diseases. For this mobile application, four actors can be identified: the User (the *main* actor) interested in getting the best route; the Green Route application (the *system* actor) interested in determining the optimal route; the FIWARE (*external* actor) interested in offering access to environmental data and publishing open data for other users; and finally, a Geographical Information System (GIS) (*external* actor) which offers geographic information. The information system (the *system* actor) must explicitly take into account these actors' intentions to determine which ones (and to what extent) will be considered in the system to be developed. Despite the syntactical similarities of a value@GRL model with a model built using the GRL, the main difference comes from the purpose of modeling, which is to represent the intentional elements as value propositions for each actor regarding the IS to be developed. This purpose constrains the use of elements, focusing only on those intentional elements that affect or are affected by the system. The links among intentional elements within an actor and among actors and the system are fundamental to propagate the value intentions among them.

The next sections detail how the value@GRL model is created and then how the values of its model elements are propagated.

2.3. Goal Modeling

The goal modeling activity consists in the identification and specification of the involved actors in the context of the IS. Besides the expert domain knowledge, this activity has the problem statement and the guidelines for modeling value in GRL as inputs. The problem statement describes what will be modeled in the value modeling (actors, intentional elements, links, etc.), according to the expert domain knowledge. The guidelines explained below show how to create a value model using value@GRL.

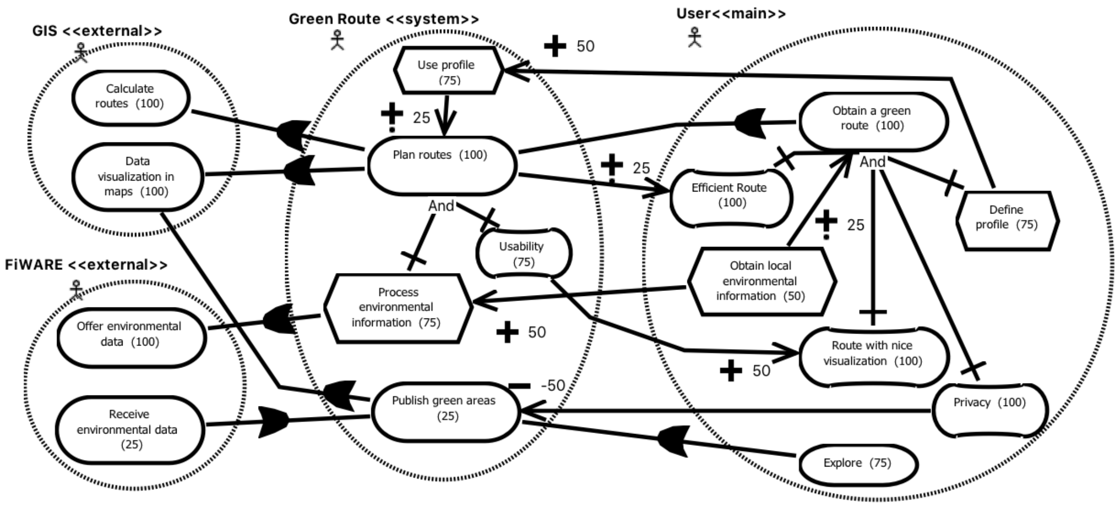

Fig. 2. Green Route modeled with value@GRL.

According to the guideline, there are steps for modeling the main actor(s) and the external one(s), and steps for modeling the system actor. Each actor' specification will include its value propositions represented by the intentional elements and relationships among them. It is important to highlight that all actors (main and externals) must have a relationship with the system actor. This guarantees that the system actor will attend the goals from different actors. The result is the value model for the IS. Thus, the goal modeling activity can be decomposed into the following detailed tasks (guidelines):

- **Task 1: Specification of actors**. Actors (main and external actors) related to the system to be developed are identified. At this moment, the system actor is not yet represented since the purpose is to analyze and reason about active actors in the domain. For the example, as shown in Figure 2, the *main actor* is the User and the *external ones* are the GIS and the FIWARE.
- **Task 2: Specification of intentional elements**. The value propositions of each actor are represented as intentional elements (goals, soft-goals, and tasks) within the actor's border. For example, according to Figure 2, the external actor FIWARE has the goals *offer environmental data* and *receive environmental data*.
- **Task 3: Specification of links among intentional elements**. Once the intentional elements are identified and specified within the actor's border, they are analyzed in order to specify the possible links among them. Two kinds of links are specified: decomposition (AND, OR, and XOR) and contribution (positive or negative). A goal can be decomposed into any intentional element. A task can be decomposed into another task or soft-goal. Finally, a soft-goal can be decomposed into another soft-goal. Only one level of decomposition is recommended. The contribution is also labeled using a quantitative scale from 0 to 100 in intervals of 25, where 0, means no contribution; 25 low contribution; 50 medium contribution; 75 high contribution, and 100, very high contribution. When specifying a contribution link for intentional elements with decomposition, only the father or the child should be selected in order to avoid both contributing to the same intentional element. As an example, in Figure 2, the main actor User has the goal *obtain a green route* decomposed (AND type) into *efficient route*, *route with nice visualization*, *privacy,* and *define profile*.
- **Task 4: Specification of the system actor**. This task consists in identifying and modeling the system actor. Figure 2 shows the modeled system actor *Green Route*.
- **Task 5: Specification of the system intentional elements**. After specifying the system actor it is necessary to identify and specify its intentional elements. Moreover, all the intentional elements from the system actor must have a link with at least one intentional element from another actor, except for children intentional elements in which the father already has a link. Thus, we can guarantee that the system actor will respond to the value

propositions from other actors. For example, in Figure 2 the system actor *Green Route* has the following intentional elements: *use profile*, *plan routes*, *usability*, *process environmental information*, and *publish green areas*.

- **Task 6: Specification of links among intentional elements of the system actor.** The links among the system actor intentional elements are identified and specified as was done in Task 3. Figure 2 shows the system actor Green Route with decomposition link (AND type) in which the goal *plan routes* is decomposed into *process environmental information* and *usability*.
- **Task 7: Specification of dependency and contribution links between the system actor and other actors.** The main difference when modeling the system actor is that dependency links also need to be specified. This means that those intentional elements from actors that depend on intentional elements of system actor (or vice-versa) are identified. Dependency links among intentional elements that do not involve the system actor are not of interest since our purpose is to reason about the capabilities of the IS to be developed. Also, it is necessary to specify contribution links between the system actor intentional elements and other intentional elements from other actors. Figure 2 presents some dependencies and contributions among the actors and the system Green Route (and vice-versa). For example, the goal *plan routes* from Green Route depends on the goal *calculate routes* and the goal *data visualization in maps* from the GIS external actor. The task *obtain local environmental information* from the User contributes positively to the task *process environmental information* from the Green Route system actor.

Figure 2 shows an output of this activity. However, the number in brackets of each intentional element is not a result of this activity. They will be specified in the next activity.

2.4. Goal Model Value Propagation

Indeed, the first task of this activity is to establish the importance of each of the modeled intentional elements, represented by the numbers in brackets in Figure 2. We consider that the intentional elements are not value-neutral for the stakeholders. This means that they will be able to understand and prioritize their own intentional elements. The importance is established in a scale from 0 to 100, with intervals of 25. Thus, the importance will indicate which intentional elements are more relevant to accomplish an actor goal. This task should be performed for all actors, starting with the main actor, then the external ones, and finally, the system actor.

The assigned importance to each intentional element is then propagated internally to the other intentional elements through the various links (contribution and dependency links) that connect them. The propagation is calculated first for the main actor(s), then the external(s) one(s), and finally for the system actor. We defined the Equation 1, which is a variant of the Weighted Sum Model (WSM), as an additive aggregation approach to calculate the propagation of the value through the links connecting the intentional elements.

$$Cal.Value(IE) = IE_{Imp} + \sum_{i=0}^{n}\left(\frac{IE_{contrib}}{100} * Value(IE_{ImpDest})\right) + \sum_{i=0}^{n}\left(Value(IE_{ImpDest})\right) + \left(\frac{IE_{father_{Imp}}}{\#IE_{siblings}}\right) \quad (1)$$

Where:

IE_{imp}: Intentional Element importance.
$IE_{contrib}$: Value of the contribution (in or out from the IE).
$IE_{ImpDest}$: Importance value of the IE at the end of the contribution/dependency.
$IE_{fatherImp}$: Importance value of the father IE from decomposition.
$\#IE_{siblings}$: Total number of siblings in decomposition.

This equation defines the calculated value of an intentional element (IE) by considering four elements: its own importance, the sum of the contributions that the IE has (contribution links), the sum of the dependencies that the IE has (dependency links), and finally, if the IE is part of greater IE (decomposition link) takes a proportional value of the importance of the compound IE, considering the number of siblings. The calculated importance is firstly defined for parent intentional elements and then for child ones. Note that, once the importance of the intentional elements within all the actors (main an externals) is determined, they are also propagated through the contribution and dependency links to the system actor, meaning that the system actor alters its own values according to the values of other involved actors. Following, the composition of each part of Equation 1 (algorithms) will be explained in detail.

2.4.1. Calculating the Sum for Contribution Links

This calculation is based on the contributions (in and out) that an intentional element has as well as on the importance of the target intentional element. For each contribution link, firstly, we measure its weight, which is calculated by dividing the contribution value between 100, resulting in four possible values 1, 3/4, 2/4, or 1/4 corresponding to the value of the contribution link 100, 75, 50, or 25. Once we have calculated the contribution weigh, we multiplied it by the importance value of the target intentional element. This must be done for all the contributions of the intentional element. The result of each multiplication is then summed. It should be noted that Equation 1 does not distinguish between the directions of the contributions. We assume that because the direction is very subjective and may change depending on the modeler. For example, in Figure 2, one can define a contribution link between usability and visualization having in mind that "usability helps to obtain a route with nice visualization" or on the other way around "a route with nice visualization contributes to usability".

2.4.2. Calculating the Sum for Dependency Links

The sum of the dependency links consists in summing the importance value of the target intentional element from the dependency link.

2.4.3. Calculating Decomposition Links

One of the main differences with the algorithm proposed by Amyot et al. [3] is that value@GRL propagates the importance from the parent element to the child, spreading it equally among the children. If the importance was propagated from the child to the parent, the parent with more children will be the parent with a higher calculated value. Thus, to avoid this problem, in our equation, a decomposition link indicates that children elements compose a parent and they will have their own importance value. Thus, children importance values may be incremented because they are part of a composite.

2.4.4. Calculated Value for Intentional Elements

After calculating the value for the main actor(s), then the external(s) one(s), and finally for the system actor, it is possible to build a ranked list by the intentional elements calculated importance. Table 1 shows an excerpt of the intentional elements (from Figure 2) for each actor together with the initial importance (Imp.) and the result after applying the propagation equation (algorithms) to calculate CV (Calculated value), which represents the value of the intentional element for a given actor. The gray shade in Table 1 corresponds to soft-goals that can be associated later to non-functional requirements.

3. Using the Value Model for Deriving a BPMN Model

The purpose of the activity *high-level business process modeling* shown in Figure 1 is to provide a general view of the high-level business process activities that are needed to realize the system to be developed. This activity uses as input the value@GRL model from the goal modeling activity. We have defined a traceability matrix, which maps elements from the value@GRL model into elements of the BPMN model, as shown in Table 2. In order to transform a value@GRL model into a BPMN model, we propose the following rules:

- **Rule 1: Actor transformation**: Actors from the value@GRL model are part of the system context; therefore, they will be transformed into pools or lanes in the BPMN model. All actors, who have a straight relationship with the system, will be included in the process model. This relationship is represented by a link (dependency or contribution) with an intentional element of the system actor.

Table 1. Intentional Element Importance and Calculated Value.

Actor	Intentional Element (IE)	Imp.	Calculated Value (CV)
User	**IE6.** Obtain a green route	100	212,50
	IE7. Define profile	75	137,50
	IE8. Obtain local environmental information	50	112,50
	IE9. Explore	75	100,00
	IE10. Route with nice visualization	100	162,50
	IE11. Efficient route	100	150,00
	IE12. Privacy	100	112,50
Green Route system	**IE1.** Plan routes	100	831,25
	IE2. Process environmental information	75	356,25
	IE3. Publish green areas	25	343,75
	IE4. Use profile	75	267,19
	IE5. Usability	75	160,94

Table 2. Traceability between value@GRL and BPMN models.

GRL Element	Cardinality Rule	BPMN Element	Rule
Actor	0..1	Pool/Lane	1
Intentional Element: Task	1..*	Process/Activity	2a
Intentional Element: Goal	0..*	Process/Activity/Event	2b
Intentional Element: Soft-Goal	0..*	Process/Activity/Event	2c
Link: Dependency	0..*	Flow	3a
Link: Contribution	0..*	Flow	3b
Link: Decomposition	0..1	Process/Flow	3c

- **Rule 2: Transformation of intentional elements**: Intentional elements from each actor of the value@GRL model will be included into the corresponding BPMN pool/lane.
  a. **Task**: A task in value@GRL indicates a specific way of doing something. Its correspondence in BPMN is an activity or a process. A task in value@GRL is described at high level therefore it can be derived to an atomic activity or a process that will be expanded later.
  b. **Goal**: A goal represents an assertion that an actor wants to get. None or several activities, processes, or even events can carry this assertion.
  c. **Soft-goal**: Similar to a goal, a soft-goal can be derived to an activity, process, or to an event. However, the characteristics of soft-goals sometimes refer to non-functional requirements, which are difficult to represent in an operational language. Thus, they may not have a derived element. Instead, they may have a concept to be taken into account throughout the process.

- **Rule 3: Links transformation**: Links can be represented as flows in BPMN. Sometimes links will not have a representation in the BPMN model.
  a. **Dependency**: It represents a flow of sequence or message. It indicates a needed relationship between two elements. A dependency link between two intentional elements from two actors indicates the existence of a flow between activities of the two pools.
  b. **Contribution**: It is a flow of sequence, message, or it may also not be represented in the BPMN model. It indicates the way in which an intentional element affects other ones.
  c. **Decomposition**: It indicates the division of an element into smaller parts. This type of link is related to processes and sub-processes (depending on the type of the children and parent intentional elements). A parent element from the decomposition will be

represented as a process. The child element of decomposition will be an activity or sub-process. Finally, child elements of decomposition will be represented as a flow between activities of the derived thread.

Figure 3 shows the resulting BPMN model for the Green Route system after applying the transformation rules discussed above. Each intentional element from the value@GRL model (see Figure 2) has a correspondent element in the BPMN model. For instance, the Green Route system actor is mapped as a pool *Green Route* in the BPMN model (Rule 1). The task *process environmental information* from the Green Route system is mapped into the process *environmental information service* in the BPMN model (Rule 2a).

3.1. Business Process Activities Prioritization

Once the calculated value@GRL and BPMN models are specified, it is possible to start prioritizing the BPMN activities. These activities need to be prioritized to identify which ones will be included in a given software development increment.

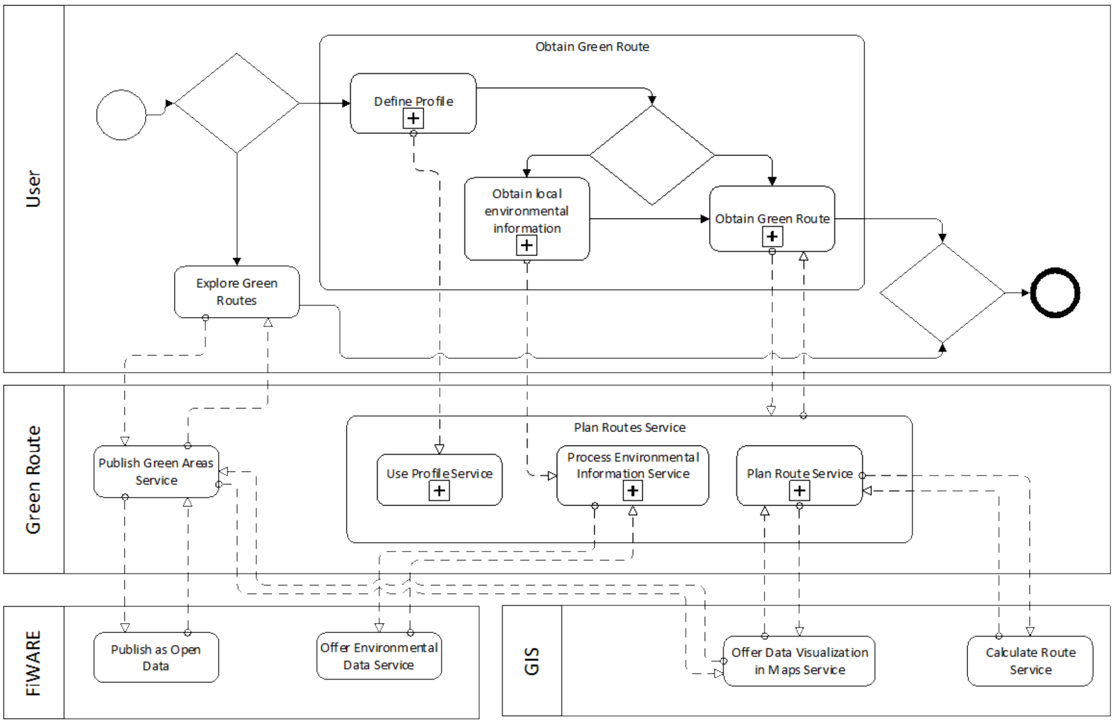

Fig. 3. BPMN model for the Green Route system.

In our approach, we enriched the prioritization process by considering the value that each activity will provide, according to the value model previously defined. To illustrate this process, Table 3 shows the calculated priority of each activity using the formula: *Priority* = $(\Sigma_{CV})/Effort$. The Σ_{CV} represents the sum of CV from IE's that satisfy a BMPN activity. Thus, these prioritized activities may drive the software development activities by choosing those activities with highest priority to be developed first. We used this strategy of defining the priority based on the effort (man-hours) and the calculated value; however, it is also possible to use other factors, as explained in the related work and conclusions sections.

In our example of the Green Route system, the three most important activities, which have a higher value and require less effort, will be: Plan Route Service, Use Profile Service, and Publish Green Areas Service. Based on this prioritization, it is possible now to define the increments that will be developed. This will be defined based on the software development team capacity. For example, if a software development team can afford 80h (effort in man-hours), this will result in three iterations, according to Table 3. The first increment will be the

Plan Route Service and Use Profile Service activities, since they sum a total effort of 75h, which is a total below the 80h (total capacity of the development team). Following the same pattern, the second increment will be the Publish Green Areas Service, with an effort of 60h, and finally, the last increment will be the Process Environmental Information Service, which has an effort of 75h.

Table 3. Prioritized BPMN Activities, according to incremental software development.

Lane	BPMN activity	Effort	IE	∑ CV	Priority = (∑CV/ Effort)	Increments
Green Route	Plan Route Service	50h	IE1	831,25	16,63	1st
	Use Profile Service	25h	IE4	267,19	10,69	
	Publish Green Areas Service	60h	IE3	343,75	5,72	2nd
	Process Environmental Information Service	75h	IE2	356,25	4,75	3rd

4. Related Work

We categorized the related work into three groups: approaches that deal with the evaluation propagation among intentional elements; proposals that deal with the downstream system development from early requirements to business processes; and requirements engineering prioritization mechanisms.

With regard to propagation techniques, Amyot et al. [3] proposed evaluation propagation algorithms to calculate the satisfaction of an intentional element based on the pre-defined user satisfaction. These algorithms firstly calculate the evaluation value for decomposition links, then for contribution links, and lastly for dependency links. They have also implemented these algorithms in a tool called JUCMNav[1]. Comparing the algorithms proposed by Amyot et al. [3] with the ones from value@GRL, only the contribution algorithm matches. The decomposition and dependencies propagations algorithms are distinct. The algorithms in value@GRL consider that values of intentional elements are not evenly propagated throughout the actors. Value@GRL includes the concept of system actor, which is the system-to-be. The system actor should satisfy the needs of the remaining GRL actors (*i.e.*, main actor and external actors). Thus, value@GRL performs propagation internally for these actors, then, with basis on this calculated propagation, it calculates the value of the system actor intentional elements. Hence, value@GRL ensures that the calculated value from others actors is propagated to the system actor. This propagation allows the intentional elements of the system actor to satisfy the intentional elements of the rest of the actors, thus improving the current propagation mechanisms. The value@GRL decomposition algorithm deals with the top-down approach in which the value from a father intentional element is proportionally propagated to its children elements. This mechanism avoids the problem of increasing too much the father value according to the number of children. Another perceived difference between the algorithms proposed by Amyot et al. [3] and the value @ GRL is related to the dependency propagation algorithm. Value@GRL uses the sum of all values in which the intentional element has a dependency link (in or out). This strategy highlights the importance of dependency links since it increases the value of an intentional element according to the number of its dependency links.

Aprajita and Mussbacher [4] proposed aggregated contributions to cope with the decomposition of intentional elements. They solve an existing problem in the Amyot et al. [3] evaluation propagation algorithm that children and father from decomposition links contribute to the same intentional element. This may lead to the increase of the value from the intentional element that is receiving the contribution from the father and its children.

[1] Tool available online at http://jucmnav.softwareengineering.ca/jucmnav/ (accessed in April, 2017).

However, in order to make the value@GRL model specification simpler, we defined guidelines to deal with this condition. If the father intentional element contributes to another intentional element, its children intentional elements do not need to contribute. Something similar happens if the children intentional elements contribute to another intentional element; the father does not need to contribute to it. Thus, through guidelines, aggregated contribution can be avoided in value@GRL.

In relation with the transformation of goal-oriented models into business process, Alves et al. [2] proposed some heuristics to obtain a BPMN model [14] from an i* model [15] and vice-versa. They claim that the integration of both models may lead to a better understanding of BPMN activities, including their execution order, and their relationship with the goal-oriented model elements to cope with the goals of the organization. Cortes-Cornax et al. [7] present a bridge, called intentional fragments, between BMPN 2.0 models and KAOS goal-oriented models. The intentional fragments make explicit the relationship between a BPMN activity and its corresponding goal in KAOS. However, these approaches do not deal with the prioritization of BPMN activities. In value@GRL, after transforming a goal-oriented model into a BMPN model through a mapping of GRL and BMPN elements, we also deal with the prioritization of BMPN activities in order to define the increments in an incremental software development. Decreus et al. [8] presented five challenges that still exist in transforming i* goal models into business process models. One of the challenges is related to the mapping between the models. We propose a traceability matrix in value@GRL to map goal-oriented model elements to BPMN elements. Thus, all elements from a goal-oriented model are mapped into at least one BMPN element.

However, these proposed approaches neither focus on transforming intentional elements into business processes according to the system actor nor focus on incremental software development, such as value@GRL does. Namely, in value@GRL, the system is going to be incrementally developed according to the value that its goals provide to other actors. According to the calculated value of intentional elements in a value@GRL model and after its transformation into a BMPN model, it is possible to prioritize the BPMN model activities according to their calculated value and effort. Thus, activities with a high priority will be the ones that bring more value to the stakeholders (*i.e.*, those with a high calculated value and small effort).

Finally, with regard to the prioritization mechanisms, value@GRL uses a combination of effort and the calculated propagated value to define the prioritization order of BMPN activities; however, other properties can also be used to prioritize. In Berander and Andrews [5], the authors provide an overview of requirements engineering prioritization techniques that use different properties, such as importance, penalty, cost, time, risk, volatility, and their combination, to prioritize requirements. Moreover, they discuss the importance of involving the stakeholders in the prioritization process.

Kaiya et al. [12] presented the Attribute Goal-Oriented Requirements Analysis (AGORA) method that helps to prioritize (ranks) goals among stakeholders. This method deals only with AND/OR decompositions and contributions in the goal-oriented modeling. Value@GRL differently deals also with dependency relationships among actors for defining the calculated value of intentional elements in order to prioritize the BMPN activities. The inclusion of the dependency link allows to capture and to calculate better values, since they will represent a dependency between actors' intentional elements with the system and vice versa. Thus, the system will attend others actors' needs according to theirs' value propositions.

5. Conclusions and Further Work

In this paper, we have presented a specialization of the GRL language to support the modeling of value regarding to the IS to be developed. This model can then be used to prioritize business processes in the context of incremental software development. The main benefit of this approach is that value is equally engineered than other high-level concepts such as processes or activities, when developing IS and also provides additional information to make informed decisions on the selection of increments, thus improving the current practices that are mainly focused on effort estimation and experience as decision criteria. An excerpt of a mobile application for smart cities has been used to show the use of our approach.

There are some limitations in prioritizing requirements as stated by Achimugu et al. [1]. One of them is the communication among stakeholders. We believe that our approach contributes in this activity by defining the importance of intentional elements using value@GRL. Thus, as a future step, we plan to perform an experiment to measure the effectiveness of subjects when assigning importance for the intentional elements, after dialoguing with all the stakeholders. Moreover, we also plan to improve the prioritization of BPMN activities. We believe that other factors such as risk, penalty, volatility, and their combination may also influence the prioritization of business processes.

References

1. Achimugu, P., Selamat, A., Ibrahim, R., Naz, M.: A systematic literature review of software requirements prioritization research. Inf. Softw. Technol., vol. 56, no. 6, pp. 568–585, 2014.
2. Alves, R., Silva, C., Castro, J.: A bi-directional mapping between i* and BPMN models in the context of business process management. CEUR Workshop Proc., vol. 1005, 2013.
3. 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., 25:841–877, 2010.
4. Aprajita, Mussbacher, G.: Aggregate Contribution of Decomposed Intentional Elements. Proceedings of the Ninth International i* Workshop, pp. 73–78, 2016.
5. Berander, P., Andrews, A.: Requirements Prioritization. In Engineering and Managing Software Requirements, A. Aurum and C. Wohlin, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 69–94, 2005.
6. Boehm, B.: Value-Based Software Engineering: Overview and Agenda. In S. Biffil, A. Aurum, B. Boehm, and P. Grunbacher, Value-Based Soft. Eng., Springer, 2006.
7. Cortes-Cornax, M., Matei, A., Letier, E., Dupuy-Chessa, S., Rieu, D.: Intentional fragments: Bridging the gap between organizational and intentional levels in business processes. Lect. Notes Comput. Sci., vol. 7565 LNCS, no. pp. 110–127, 2012.
8. Decreus, K., Snoeck, M., Poels, G.: Practical challenges for methods transforming i* goal models into business process models. Proc. IEEE Int. Conf. Req. Eng., pp. 15–23, 2009.
9. Estrada, H., Najera, K., Vázquez, B., Martínez, A., Téllez, J. C., Hierro, J. J.: Applying Tropos Modeling for Smart Mobility App Based on the FIWARE Platform. In iStar workshop, 85-90, 2016.
10. Horkoff, J., Li, F. L., Salnitri, M., Cardoso, E., Giorgini, P., Mylopoulos, J.: Using goal models downstream: A systematic roadmap and literature review. Int. Journal of Information System Modeling and Design (IJISMD) 6(2), 1-42, 2015.
11. ITU-T. Recommendation Z.151 (09/08): User Requirements Notation (URN)– Language definition, 2008.
12. Kaiya, H., Horai, H., Saeki, M.,: AGORA: Attributed goal-oriented requirements analysis method. IEEE Int. Conf. Requir. Eng., vol. 2002–January, pp. 13–22, 2002.
13. Khurum, M., Gorschek, T., Wilson, M.: The Software Value Map – an Exhaustive Collection of Value Aspects for the Development of Software Intensive Products. Journal of Software: Evolution and Process 25: 711-741, 2013.
14. Miers, D., Stephen, W.: BPMN Modeling and Reference Guide. Fut. Strategies Inc., 2008.
15. Yu, E.: Towards Modelling and Reasoning Support for Early-phase Requirements Engineering. 3rd IEEE Int. Symp. on Requirements Engineering, pp. 226–235, 1997.