Reasoning about Stakeholder Groups for Requirements Negotiation based on Power Relationships

Hui Yang
State Key Lab of Software Engineering
School of Computer, Wuhan University
0209yanghui@gmail.com

Peng Liang †
State Key Lab of Software Engineering
School of Computer, Wuhan University
liangp@whu.edu.cn

Abstract—With the increasing complexity and uncertainty in requirements engineering (RE), the impact of power relationships between stakeholders becomes critical to the success of requirements engineering process, especially in requirements negotiation to resolve conflicting requirements. In this paper, we make use of the basic principles of requirements negotiation and define reasoning rules to reason about stakeholder groups for requirements negotiation, based on an extended i* modeling framework that represents the power relationships between stakeholders in RE. We derive decision-makers and supporter groups of conflicting requirements using the reasoning rules based on power relationships between stakeholders, and further reason about representative stakeholders from the supporter groups to participate in requirements negotiation activity. We describe the reasoning process and provide a concrete case of reasoning about stakeholder groups for requirements negotiation based on the extended i* model.

Keywords—requirements negotiation; stakeholder group reasoning; i*; power relationships; SWRL

I. INTRODUCTION

Stakeholders in requirements engineering inevitably have different views, interests, and opinions to a system to be developed, and their proposed requirements might be conflicting to each other. The objective of requirements negotiation activity is trying to reconcile conflicting views and end up with a consistent set of requirements [17]. The conflicting requirements to be resolved in requirements negotiation is mainly due to the diverse interests and motivations from various stakeholders, who may propose, support, or object specific requirements [14]. In this situation, the conflicts between requirements from various stakeholders are inevitable. Some essential difficulties in agreeing requirements exist, that are further compounded by a number of contextual issues, including the political and social environment issues [19]. Meanwhile, software systems become increasingly complex with a large number of uncertain stakeholders, which results in complexity and uncertainty in requirements engineering (RE) consequently [4]. Improving requirements modeling methods, supporting techniques and tools is not enough to deal with the increasing complexity in the RE process, especially in requirements negotiation [12]. The social environment and organization of the whole system should be considered, especially the impact of the power relationships between stakeholders becomes critical to the success of RE process. Recent research shows that for many failed projects, negligence of the political and power relationships in RE resulted in their failure [8]. Yu considered that, as software systems become increasingly complex and densely intertwined with human social environment, a requirements model, which can represent the social characteristics of complex systems, is needed [6]. To this end, i* model is proposed as a framework for modelling and reasoning about organizational environments in RE [7]. However, one limitation of i* modeling framework is that it does not consider the power relationships between stakeholders [10].

Although there has been relatively few work that takes the influence of the power relationships in RE into account, we cannot ignore that the power relationships are an important factor determining the success or failure of a project [8]. It is also argued that power and politics should not be regarded solely as negative factors to be minimized or avoided. On the contrary, they can be viewed as tools that, if harnessed correctly, could enable the delivery of better requirements and more successful solutions [4]. Therefore we propose to reason about stakeholder groups for requirements negotiation, based on the power relationships between stakeholders using reasoning rules, and to assist requirements negotiation activity with the reasoning results.

In this work, it is assumed that the information about conflicting requirements is already known in advance (e.g., requirement A is conflicting with requirement B). We try to reason about the stakeholder groups that are involved in the requirements negotiation of these conflicting requirements. Following concrete steps are conducted in this work: we construct an ontology for stakeholder group reasoning using Protégé1, describe the reasoning rules using Semantic Web Rule Language (SWRL) [15], and reason about stakeholder groups for requirements negotiation using Pallet2 reasoning engine based on the power relationships between stakeholders represented in an extended i* model. We can then derive decision-makers of conflicting requirements and supporter groups for each conflicting requirement, considering the effect of the power relationships between stakeholders, and subsequently infer the representative stakeholders from the supporter groups, who can participate in requirements negotiation. The objective of this work is to use the reasoning results of various stakeholder groups to support requirements

† Corresponding author

1 http://protege.stanford.edu/ (version 4.3)
2 http://clarkparsia.com/pellet/protege/ (version 1.0)
II. TYPES OF POWER RELATIONSHIPS BETWEEN STAKEHOLDERS

Power, which is owned by individuals, can be viewed as the potential relationships with other individuals in a social environment. If an individual $X$ has a power relationship with individual $Y$, then one or both of the following situations will be true [4]:

1. $X$ can assign $Y$ to do something which $Y$ itself is reluctant to do, or $X$ will hinder $Y$ doing something $Y$ itself wants to do.
2. $X$ can define reality in such a way that $Y$ will perform in accordance with it.

There are many theories describing the power relationships between stakeholders. One of the most commonly-used theories is proposed by French & Raven on the base of social power, which formally defines and classifies five types of power relationships [3][4], as detailed below:

- **Reward**: the ability to bestow rewards. For example, a department head can reward an employee.
- **Coercive**: the ability to punish. For example, a department head has the power to coerce an employee.
- **Expert**: possession of skills or knowledge. For example, an employee who has special skills or knowledge may have the expert power over the department head.
- **Referent** from personal characteristics. For example, a person would like to refer the opinions of the authorities, who has personal characteristics in the field she/he respects, when making decisions.

In RE, many frameworks and techniques have been developed to help refine requirements statements to achieve better precision, completeness, and consistency. However, to develop systems that will truly meet the real needs of an organization, one often needs to have a deep understanding of how the system is embedded in the organizational environment [7], where the relationships between stakeholders matter. To this end, $i^*$ is proposed as a goal-oriented and agent-oriented requirements modeling language, which highlights the position of social actors. Actor in $i^*$ modeling framework represents an entity that is considered to have the following intention attributes: goals, beliefs, abilities, and commitments [6]. Actors can achieve an appropriate objective by means of actions. Stakeholders in requirements engineering are an important source of actors. $i^*$ model consists of two main modeling components: Strategic Dependency model (SD) and Strategic Rationale model (SR) [7]. SD model describes the dependency relationships between the important actors in the future system, and SR model is used to describe stakeholder interests and concerns, and how the actors depend on each other to achieve mutual goals and soft goals [7]. $i^*$ model is mainly used to describe and analyze how to achieve the objectives for actors by using dependency relationships, i.e., SD model and SR model, and how to reorganize the dependency relationships among actors to help actors to achieve their objectives [6][7]. However, existing $i^*$ modeling framework does not consider the power relationships between actors [10].

Milne & Maiden proposed a framework of studying power relationships in RE, based on the power structure and its political influence on decision-making process [10]. They argued that modeling on power relationships in RE could be considered as an effective tool to aid decision-making in RE. They extended the framework of studying power relationships in RE to $i^*$ model framework, based on the five types of power relationships proposed in [3]. Finally they concluded that the extended $i^*$ model can be used to describe power and politics in RE.

Hence, we propose to reason about stakeholder groups for requirements negotiation, based on the power relationships between stakeholders that the extended $i^*$ model describes. The reasoning process is described in detail in Section III.

III. REASONING STAKEHOLDER GROUPS FOR REQUIREMENTS NEGOTIATION

It is necessary and critical to include stakeholders as comprehensive as possible, who have the interest to participate in requirements negotiation [14], but the communication cost between stakeholders will be increased exponentially when the number of participated stakeholders increases. So seeking for a group of representative stakeholders who truly represent the interests of the involved stakeholders (including e.g., users, customers, and developers [1]) will considerably alleviate the communication effort in requirements negotiation. The following two aspects should be paid great attention in requirements negotiation activity:

1. Participants of requirements negotiation are supposed to cover comprehensive stakeholders, including users, customers, and developers [1].
2. Participants of requirements negotiation are supposed to be representative [2].

According to the characteristics of requirements negotiation [2], it can be concluded that representative stakeholders participating in requirements negotiation should satisfy following criteria: she/he has either a power relationship of **Expert** or **Referent** to other stakeholders. Milne & Maiden thought that the second dimension of power describes the ability of the power to decide which issues are decided upon [4]. Hence, decision-makers participating in requirements negotiation should satisfy following criteria: she/he has a power relationship of **Expert** or **Legitimate** to...
other stakeholders, and also she/he should be on the top of the power relationships (i.e., no other stakeholders have a power relationship of Expert or Legitimate to her/him).

Firstly, we should collect the information on conflicting requirements, without considering the power relationships between stakeholders, (Suppose that the information on the conflicting requirements can be obtained by making use of a framework for knowledge-based requirements engineering (KBRE) [13], and it is assumed that the conflicting requirements are \( \text{req}_1 \) and \( \text{req}_2 \)). In this step, we can obtain the following information: all the stakeholders in relation to these conflicting requirements, for instance, stakeholders who propose these conflicting requirements respectively. Secondly, we use Protégé to construct the ontology (including the definition of classes and object properties, and reasoning rules) to support the reasoning about stakeholder groups for requirements negotiation. Semantic Web Rule Language (SWRL) is used to describe the reasoning rules [15], and reasoning engine Pallet is used to reason about stakeholder groups for requirements negotiation. Finally, the reasoning results are: decision-makers of conflicting requirements and supporter group for each conflicting requirement, and the representative stakeholders subsequently inferred from the supporter groups. The whole process of reasoning about stakeholder groups for requirements negotiation is shown in Fig. 1. The reasoning rules, as a special type of requirements knowledge (RK), and the reasoning activities, are highlighted in brown in the figure.

The reasoning results of stakeholder groups obtained (decision-makers, supporter groups, and representative stakeholders) provide project managers reference information to organize and coordinate requirements negotiation more efficiently and cost-effectively. The detail information about the ontology for stakeholder group reasoning in requirements negotiation, the reasoning rules based on the ontology, as well as a case of using the reasoning process will be presented in Section IV and Section V.

IV. ONTOLOGY AND RULES OF REASONING STAKEHOLDER GROUPS FOR REQUIREMENTS NEGOTIATION

In this work, Protégé is used to construct the ontology for stakeholder group reasoning in requirements negotiation. The reasoning ontology is mainly composed of classes, object properties, and reasoning rules. Note that the reasoning rules, as a type of RK, can be changed, removed, and added according the context of specific projects. Reasoning about stakeholder groups for requirements negotiation therefore can be automatically executed based on this reasoning ontology.

A. Classes and Object Properties of Ontology for Reasoning Stakeholder Groups in Requirements Negotiation

The reasoning ontology contains two major classes (concepts) in requirements engineering, i.e., Stakeholder and Requirement. Stakeholder has three subclasses (i.e., basic types of stakeholders in software development): Developer, Customer, and User, as shown in the Protégé Class hierarchy panel in Fig. 2.

Fig. 2. Ontology classes for reasoning about stakeholder groups in requirements negotiation

Object properties of these classes in the reasoning ontology are described as follows.

- Object property hasConcern\((s_i, r)\) expresses that the Stakeholder \( s_i \) is concerned with Requirement \( r \).
- Object property hasProposed\((s, r)\) expresses that Stakeholder \( s \) proposes Requirement \( r \).
- Object property hasPowerofExpert\((s_i, s_j)\) expresses that Stakeholder \( s_i \) has a power relationship of Expert to another Stakeholder \( s_j \).
- Object property hasPowerofReferent\((s_i, s_j)\) expresses that Stakeholder \( s_i \) has a power relationship of Referent to another Stakeholder \( s_j \).
- Object property hasPowerofReward\((s_i, s_j)\) expresses that Stakeholder \( s_i \) has a power relationship of Reward to another Stakeholder \( s_j \).
- Object property hasPowerofLegitimate\((s_i, s_j)\) expresses that Stakeholder \( s_i \) has a power relationship of Legitimate to another Stakeholder \( s_j \).
- Object property isRepresentativeUserof\((s, r)\) expresses that Stakeholder \( s \) is a representative user who agrees on Requirement \( r \).
- Object property isRepresentativeCustomerof\((s, r)\) expresses that Stakeholder \( s \) is a representative customer who agrees on Requirement \( r \).

Fig. 1. Process of reasoning about stakeholder groups for requirements negotiation
- Object property isRepresentativeDeveloperof(s, r) expresses that Stakeholder s is a representative developer who agrees on Requirement r.
- Object property isDecisionMakerofRequirement(s, r) expresses that Stakeholder s is a decision-maker of Requirement r.
- Object property isnotDecisionMakerofRequirement(s, r) expresses that Stakeholder s is not a decision-maker of Requirement r.
- Object property hasConflict(r1, r2) expresses that Requirement r1 and r2 are in conflict, and hasConflict(r1, r2) implies that hasConflict(r2, r1).
- Object property agreeRequirement(s, r) expresses that Stakeholder s agrees on Requirement r.

The object properties described above are all shown in the Object property hierarchy panel in Fig. 3 with a detailed example of object property agreeRequirement (stakeholder, requirement), which “Domain” and “Range” are shown in the right part of the figure. The “Domain” field contains class Stakeholder, and the “Range” field contains class Requirement, which can be interpreted as: Stakeholder has an agreeRequirement relationship (object property) to class Requirement.

![Object Property Hierarchy Panel](image)

Fig. 3. Ontology object properties for reasoning stakeholder groups in requirements negotiation

### B. Rules of Reasoning about Stakeholder Groups for Requirements Negotiation

- **Rules of reasoning about supporter groups for requirements negotiation**

  It is obvious that stakeholders tend to agree on the requirements that are proposed by themselves. This rule is described by the reasoning rule (1) in TABLE I.

  Object property hasPowerofExpert(si, sj) implies that stakeholder sj will agree on the same requirement that stakeholder si agrees, based on the impact of Expert power relationship, likewise, object property hasPowerofReferent(si, sj) implies that the attitude of stakeholder sj to one requirement will refer to the attitude of stakeholder si to the same requirement. Hence, it can be concluded that stakeholder sj will agree on the requirements that stakeholder si agrees. Similarly, object property hasPowerofReward(si, sj) implies that if stakeholder si agrees one requirement, then under the impact of Reward power relationship, stakeholder sj will agree on the same requirement too. These rules are specified by the reasoning rules (2) (3) (4) in TABLE I.

<table>
<thead>
<tr>
<th>ID</th>
<th>Reasoning rules</th>
</tr>
</thead>
<tbody>
<tr>
<td>(1)</td>
<td>hasProposed(?,r1) -&gt; agreeRequirement(?,r1)</td>
</tr>
<tr>
<td>(2)</td>
<td>Requirement(?r),Stakeholder(?si),Stakeholder(?sj),Stakeholder(?s1),Stakeholder(?s2), hasConcern(?r,?si),hasConcern(?r,?sj),hasConcern(?r,?s1),hasConcern(?r,?s2), agreeRequirement(?si,?r) -&gt; agreeRequirement(?sj,?r)</td>
</tr>
<tr>
<td>(3)</td>
<td>Stakeholder(?si),Stakeholder(?s1),Stakeholder(?s2), Requirement(?r),hasPowerofReward(?r),hasPowerofReferent(?r), hasConcern(?r,?si),hasConcern(?r,?s1),hasConcern(?r,?s2), agreeRequirement(?s1,?r) -&gt; agreeRequirement(?s2,?r)</td>
</tr>
<tr>
<td>(4)</td>
<td>Requirement(?r),Stakeholder(?si),Stakeholder(?s1),Stakeholder(?s2), hasConcern(?r,?si),hasConcern(?r,?s1),hasConcern(?r,?s2), agreeRequirement(?s1,?r) -&gt; agreeRequirement(?s2,?r)</td>
</tr>
</tbody>
</table>

- **Rules of reasoning about representative stakeholders for requirements negotiation**

  According to the discussion above, it can be concluded that if a user A has a power relationship of Referent or Expert to another user B, and both of them agree on the same requirement, then the user A will be considered as a representative user to participate in requirements negotiation. In a similar situation, if a customer A or developer A has a power relationship of Referent or Expert to another customer B or developer B, and both of them agree on the same requirement, then the customer A or developer A will be considered as a representative customer or developer to participate in requirements negotiation. These rules are represented by the reasoning rules (5) (6) (7) (8) (9) (10) in TABLE II.

<table>
<thead>
<tr>
<th>ID</th>
<th>Reasoning rules</th>
</tr>
</thead>
<tbody>
<tr>
<td>(5)</td>
<td>User(?si1),User(?si2), agreeRequirement(?si1,?r), agreeRequirement(?si2,?r), hasPowerofReferent(?si1,?si2) -&gt; isRepresentativeUserof(?si1,?r)</td>
</tr>
<tr>
<td>(6)</td>
<td>User(?si1),User(?si2), agreeRequirement(?si1,?r), agreeRequirement(?si2,?r), hasPowerofReferent(?si1,?si2) -&gt; isRepresentativeUserof(?si1,?r)</td>
</tr>
<tr>
<td>(7)</td>
<td>Customer(?si1),Customer(?si2), agreeRequirement(?si1,?r), agreeRequirement(?si2,?r), hasPowerofReferent(?si1,?si2) -&gt; isRepresentativeCustomerof(?si1,?r)</td>
</tr>
<tr>
<td>(8)</td>
<td>Customer(?si1),Customer(?si2), agreeRequirement(?si1,?r), agreeRequirement(?si2,?r), hasPowerofReferent(?si1,?si2) -&gt; isRepresentativeCustomerof(?si1,?r)</td>
</tr>
</tbody>
</table>
• Rules of reasoning about decision-makers for requirements negotiation

If a stakeholder $A$ has a power relationship of Legitimate or Expert to another stakeholder $B$, and both of them are concerned with the same requirement, then $A$ will be regarded as a potential decision-maker of the requirement. We further reason about decision-makers over potential decision-makers. If a stakeholder $A$ and another stakeholder $B$ are both potential decision-makers, and $A$ has a power relationship of Legitimate or Expert to $B$. Since decision-makers are generally at the top of the power relationships, we then remove $B$ from the potential decision-makers, and retain $A$ as a potential decision-maker. This reasoning will be iterated until no power relationship of Legitimate or Expert can be identified between potential decision-makers. These rules are described by the reasoning rules (11) (12) (13) (14) in TABLE III.

TABLE III. RULES OF REASONING ABOUT DECISION-MAKERS FOR REQUIREMENTS NEGOTIATION

<table>
<thead>
<tr>
<th>ID</th>
<th>Reasoning rules</th>
</tr>
</thead>
<tbody>
<tr>
<td>(9)</td>
<td>Developer(?s1),Developer(?s1), agreeRequirement(?s1,?r), hasPowerofReferent(?s1,?s2) -\rightarrow isRepresentativeDeveloperof(?s1,?r)</td>
</tr>
<tr>
<td>(10)</td>
<td>Developer(?s1),Developer(?s1), agreeRequirement(?s1,?r), hasPowerofExpert(?s1,?s2) -\rightarrow isRepresentativeDeveloperof(?s1,?r)</td>
</tr>
</tbody>
</table>

A. Known Conditions for Reasoning about Stakeholder Groups in Requirements Negotiation

The known conditions, as shown in Fig. 4, are as follows: stakeholders who are concerned with the conflicting requirements, requirement1 and requirement2, including customers Jack, Ella, Alex, and Jim, and users Rose, Cathy, Henry, and Tom. Cathy proposes requirement1; Cathy has a power relationship of Referent to Rose; Tom has a power relationship of Legitimate to Cathy; Jim has a power relationship of Legitimate to Ella; Ella proposes requirement2, which is conflicting with requirement1; Ella has a power relationship of Expert to Jack; Cathy has a power relationship of Coercive to Henry; and Jack has a power relationship of Reward to Alex.

![Fig. 4 Known conditions of the stakeholders power relationships in the case](image)

B. Process of Reasoning about Stakeholder Groups for Requirements Negotiation

Step 1: Derive supporter group of the first conflicting requirement

The following reasoning is executed according to the reasoning rule (1), along with the known condition: Ella proposes requirement2:

- hasProposed(Ella,requirement2) -\rightarrow agreeRequirement(Ella,requirement2)

Then, the following reasoning is executed according to the reasoning rules (3) (4), along with previous reasoning result: agreeRequirement(Ella,requirement2) and the known conditions: Ella has a power relationship of Expert to Jack; Jack has a power relationship of Reward to Alex; Ella and Jack are stakeholders who are concerned with requirement2.

- Stakeholder(Ella),Stakeholder(Jack),Requirement(requirement2),hasConcern(Ella,requirement2),hasConcern(Jack,requirement2),hasPowerofReferent(Ella,Jack), -\rightarrow agreeRequirement(Jack,requirement2)
- Requirement(requirement2),Stakeholder(Jack),Stakeholder(Alex),hasConcern(Jack,requirement2),hasConcern(Alex,requirement2),hasPowerofReward(Jack,Alex), -\rightarrow agreeRequirement(Alex,requirement2)

Reasoning results of Step 1 are:

- agreeRequirement(Jack,requirement2)

V. A CASE OF REASONING ABOUT STAKEHOLDER GROUPS FOR REQUIREMENTS NEGOTIATION

In this section, we describe a concrete case of reasoning about stakeholder groups for requirements negotiation. The conflicting requirements are: requirement1 and requirement2, and we try to reason about the stakeholder groups for these two requirements in requirements negotiation. We first reason about decision-makers and supporter groups of conflicting requirements requirement1 and requirement2, considering the power relationships between stakeholders, and subsequently infer representative stakeholders from the supporter groups, who can participate in requirements negotiation activity.
That is to say, the supporters of the first conflicting requirement \texttt{requirement2} are Jack, Alex, and Ella. The reasoning results about supporter group of \texttt{requirement2} are showed in the red rectangular area in Fig. 5, which is a snapshot of the DL Query tool (a plug-in of Protégé) for the semantic query [20]. The query returns all the stakeholder instances that supporting \texttt{requirement2} as the reasoning results. Note that, we only provide text descriptions of the reasoning results in remaining steps without tool snapshots, due to the space limitation.

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{fig5.png}
\caption{Fig. 5. Reasoning results of the supporter group of \texttt{requirement2}}
\end{figure}

**Step 2: Derive supporter group of the second conflicting requirement**

The following reasoning is executed according to the reasoning rules (1) (2), along with the known conditions: Cathy proposes \texttt{requirement1}; Cathy has a power relationship of Referent to Rose; \texttt{requirement2} and \texttt{requirement1} are in conflict.

- \texttt{hasProposed(Cathy,requirement1)}
- \texttt{agreeRequirement(Cathy,requirement1)}
- \texttt{requirement(requirement1),Stakeholder(Cathy),Stakeholder(Rose),hasConcern(Cathy,requirement1),hasConcern(Rose,requirement1),hasPowerofReferent(Cathy,Rose),agreeRequirement(Cathy,requirement1)}
- \texttt{agreeRequirement(Rose,requirement1)}

Reasoning results of **Step 2** are:

- \texttt{agreeRequirement(Cathy,requirement1)}
- \texttt{agreeRequirement(Rose,requirement1)}

**Step 3: Derive representative stakeholders from supporter group of the first conflicting requirement**

The following reasoning is executed according to the reasoning rule (8), along with the reasoning results of **Step 1**: the supporters of the first conflicting requirement \texttt{requirement2} are Jack, Alex, and Ella, as well as the known conditions: Ella has a power relationship of Expert to Jack; customers Ella and Jack.

- \texttt{Customer(Ella),Customer(Jack),agreeRequirement(Ella,requirement2),agreeRequirement(Jack,requirement2),hasPowerofExpert(Ella,Jack) <= isRepresentativeCustomerof(Ella,requirement2)}

Reasoning result of **Step 3** is:

- \texttt{isRepresentativeCustomerof(Ella,requirement2)}

**Step 4: Derive representative stakeholders from supporter group of the second conflicting requirement**

The following reasoning is executed according to the reasoning rule (5), along with the reasoning results of **Step 2**: the supporters of the second conflicting requirement \texttt{requirement1} are Cathy and Rose, as well as the known conditions: Cathy has a power relationship of Referent to Rose; users Rose and Cathy.

- \texttt{User(Rose),User(Cathy),agreeRequirement(Cathy,requirement1),agreeRequirement(Rose,requirement1),hasPowerofReferent(Cathy,Rose) <= isRepresentativeUserof(Cathy,requirement1)}

Reasoning result of **Step 4** is:

- \texttt{isRepresentativeUserof(Cathy,requirement1)}

**Step 5: Derive decision-makers of conflicting requirements**

The following reasoning is executed according to the reasoning rules (11) (12), along with the known conditions: Tom has a power relationship of \texttt{Legitimate} to Cathy; Jim has a power relationship of \texttt{Legitimate} to Ella.

- \texttt{Stakeholder(Tom),Stakeholder(Cathy),requirement(requirement1),hasConcern(Cathy,requirement1),hasConcern(Tom,requirement1),hasPowerofLegitimate(Tom,Cathy) <= isDecisionMakerofRequirement(Tom,requirement1)}
- \texttt{Stakeholder(Jim),Stakeholder(Ella),requirement(requirement2),hasConcern(Jim,requirement2),hasConcern(Ella,requirement2),hasPowerofLegitimate(Jim,Ella) <= isDecisionMakerofRequirement(Jim,requirement2)}

Reasoning results of **Step 5** are:

- \texttt{isDecisionMakerofRequirement(Tom,requirement1)}
- \texttt{isDecisionMakerofRequirement(Jim,requirement2)}

The DL Query plugin in Protégé is a tool used for querying the reasoning results, which is further described with following queries used in the above steps: “Stakeholder and \texttt{(agreeRequirement value requirement1)}” is used for querying the Stakeholder instances that support \texttt{requirement1}, and the query results are Rose and Cathy. “Stakeholder and \texttt{(agreeRequirement value requirement2)}” is used for querying the Stakeholder instances that support \texttt{requirement2}, and the query results are Ella, Jack, and Alex. “Customer and \texttt{(isRepresentativeCustomerof value requirement2)}” is used for querying the representative Customer instances that support \texttt{requirement2}, and the query result is Ella. “User and \texttt{(isRepresentativeUserof value requirement1)}” is
used for querying the representative user instances that support requirement1, and the query result is Cathy. “Stakeholder and (isDecisionMakerOfRequirement value requirement1) and not (isNotDecisionMakerOfRequirement value requirement1)” is used for querying the instances of decision-maker on requirement1, and the query result is Tom. “Stakeholder and (isDecisionMakerOfRequirement value requirement2) and not (isNotDecisionMakerOfRequirement value requirement2)” is used for querying the instances of decision-maker on requirement2, and the query result is Jim.

C. Using Reasoning Results to Assist Requirements Negotiation

The conflicts of goals (e.g., the conflicts arising from system functional and non-functional requirements) should be resolved through negotiation between stakeholders without necessarily weakening satisfaction of each stakeholder’s goals [19]. During requirements negotiation, it is very important that all the involved stakeholders can have equal opportunities to participate in the negotiation activity and express their opinions and interests, or there are representative stakeholders who can do so on behalf of them [16]. It is also critical to avoid such situation when key stakeholders feel that their interests are not being considered, and they believe that the requirements to be implemented may negatively impact their rights and status, etc., which may lead to the failure of the entire system. Hence, it is advisable that participants in requirements negotiation should include representative stakeholders from users, customers, developers, and many others depending on the context of the projects.

Using the reasoning results presented in Section V-B, we can derive decision-makers and supporter groups of conflicting requirements, as well as representative stakeholders from the supporter groups to participate in requirements negotiation activity. In this way, we can guarantee that representative stakeholders are involved in the discussion for requirements negotiation.

During the process of negotiating among stakeholders over the conflicting requirements, a cost-value analysis on the conflicting requirements can be conducted, i.e., the cost to implement the requirements compared to the value of the implemented requirements to the system [21]. The cost-value analysis results are available to all the stakeholders who participate in negotiation, and stakeholders are allowed to vote on the requirements [1]. Decision-makers derived from the reasoning results can then decide whether requirements should be included, excluded, or refined based on the votes and their understanding.

According to the reasoning results of the case provided in this section, the supporter group of requirement2 is composed of Jack, Alex, and Ella; the supporter group of requirement1 contains Rose and Cathy; representative stakeholder agreeing on requirement1 is Cathy; representative stakeholder agreeing on requirement2 is Ella; Tom and Jim are the decision-makers of the conflicting requirements. During requirements negotiation, Ella and Cathy as representative stakeholders are encouraged to negotiate over conflicting requirements, but the final decision on requirement1 and requirement2 is made by Tom and Jim.

VI. RELATED WORK

Some research work also took the power relationships between stakeholders in RE into consideration [9][10]. Maiden emphasized the inevitability of power and politics factors in RE, and proposed that power and politics should be explicitly represented, by e.g., extending i* model or other social modeling methods [5]. Milne & Maiden reviewed the power and politics culture in requirements engineering and discussed the impact of power and politics on decision-making process in RE, as the system organizational environment becomes increasingly complex [10]. They proposed a framework for the study of power and politics in RE, which can be used to describe, analyze, and diagnose power and politics factors in RE, and they also discussed the process of decision-making in RE based on power and politics in the proposed framework [10]. Requirements emerge in a highly collaborative, interactive, and interdisciplinary negotiation process that involves heterogeneous stakeholders [2]. In order to eliminate the conflicts between requirements, stakeholders involved in conflicting requirements should negotiate, in order to reach a compromised solution which can be accepted by all concerned stakeholders [11]. During requirements negotiation, it is essential and critical that the interests and opinions of all the involved stakeholders of conflicting requirements should be considered [16]. The WinWin negotiation model provides a framework to specify the conflicting requirements and further harmonize conflicting requirements in an iterative way [1]. It can provide a relatively acceptable and compromised solution for all the concerned stakeholders. The WinWin negotiation model successfully supports general requirements negotiation. However, making decisions when stakeholders evaluate alternative requirements is still an ad-hoc process. To this end, In et al. proposed a requirements negotiation model based on multi-criteria analysis, MPARN (Multi-Criteria Preference Analysis Requirements Negotiation) [12]. The MPARN model cooperates with the artifacts of the WinWin analysis (e.g., win conditions, issues, and options) and produces systematically negotiated agreements using multi-criteria preference analysis techniques.

In this work, we make use of the basic principles of requirements negotiation introduced in [14] and [16], and reason about stakeholder groups for requirements negotiation, based on the power relationships between stakeholders represented in the extended i* model, in order to assist requirements negotiation.

VII. CONCLUSIONS AND FURTHER WORK

Stakeholder group identification is important in requirements negotiation activity. We intend to reason about stakeholder groups for requirements negotiation, considering the impact of the power relationship between stakeholders. Technically, we use Protégé to construct the ontology for stakeholder group reasoning based on the power relationships between stakeholders represented in the extended i* model, use SWRL to describe the reasoning rules, and reason about stakeholder groups for requirements negotiation using Pallet reasoning engine. In this way, we can derive decision-makers and supporter groups of conflicting requirements, and subsequently infer representative stakeholders from the
supporter groups to participate in requirements negotiation activity. Moreover, we introduce the reasoning process in steps and provide a concrete case of reasoning about stakeholder groups to demonstrate the application of the approach in requirements negotiation activity.

We also recognize that, in a real project, the power balance and relationships between stakeholders may be changed according to the project progress. In such situation, the known conditions for stakeholder group reasoning can be modified accordingly, and then re-perform the reasoning to get updated results, which is an advantage of our reasoning method, and effective for the future update of requirements knowledge (known conditions as well as the reasoning rules) [23].

In the next step, the proposed approach can be maturated in two aspects: (1) the ontology classes, object properties, and reasoning rules can be extended, enriched, and adapted according to specific context of requirements negotiation; (2) empirically validating the effectiveness of reasoning about stakeholder groups for requirements negotiation, e.g., using controlled experiments in an industrial setting.

We also outline the promising future work directions in following points: (1) How to detect and deal with the conflicts or cycles in power relationships between stakeholders, to better reason about stakeholder groups for requirements negotiation. (2) How to (re)use the known conditions, reasoning results, and reasoning rules, as requirements rationale knowledge, to support other RE activities [23], e.g., understand the priority of requirements, detect other potential conflicting requirements in requirements analysis with collective knowledge [24]. (3) It is often problematic that one stakeholder has much strong power than other stakeholders and the stakeholder may force unreasonable requirements to other weak stakeholders. Then how to detect potential powerful stakeholders and their unreasonable requirements to avoid such situations. (4) Combine stakeholder group reasoning for requirements negotiation based on the power relationships with classical WinWin requirements negotiation model; for example, recommend representative stakeholders and decision-makers through the power relationships between stakeholders, for WinWin requirements negotiation [1]. (5) Provide integrated tool support for stakeholder group reasoning in requirements negotiation, e.g., using semantic wikis [18][22].

ACKNOWLEDGMENT

This work has been partially sponsored by the NSFC under Grant No. 61170025 and the Programme of Introducing Talents of Discipline to Universities under Grant No. B07037. We would also like to thank the anonymous reviewers for their constructive comments on this paper.

REFERENCES