AMERICAN SCIENTIFIC PUBLISHERS

Advanced Science Letters
Volume 7, Pages 135–139 (30 March 2012)

Service Discovery and Recommendation in Rough Hierarchical Granular Space

Liang Hu[1], Jian Cao[1,*], and Zhiping Gu[2]

[1]*Shanghai Key Laboratory of Scalable Computing and Systems, Department of Computer Science and Engineering, Shanghai Jiaotong University, China*
[2]*Department of Electrical Engineering Shanghai Technical Institute of Electronics and Information, China*

Service discovery is one of the most vital components involved in almost all service applications. Many researches have been contributed to improve the matching accuracy. However, without a clarified requirement specification to describe the goals that users really expect to achieve, any matching algorithm is ineffective. Goal-oriented requirement engineering is a formal requirement analysis methodology which recursively decomposes a complex requirement into a set of finer grained goals. Such a hierarchical granulation structure partitions a requirement into a family of fine grain-sized granules. Furthermore, to handle uncertainties, rough set theory is employed in granular computing. For any given imprecise user requirement, a set of ordered stratified rough set approximations can be induced over all possible partitions. These approximations are used to iteratively refine imprecise requirement and recommend goals most probably desired. We also demonstrate a case to prove that rough set theory combining with granular computing is powerful to handle imprecise requirements so as to provide better service quality.

Keywords: Requirement Engineering, Goal Model, Granular Computing, Rough Set.

1. INTRODUCTION

The service-oriented architecture (SOA) has become the first considered technology to reuse and integrate all kinds of heterogeneous applications. One of the most vital issues in almost all of service applications is the service discovery, so that some standard interfaces such as UDDI have been proposed to cover such area. Recently, with the upsurge in semantic technology, the keyword based discovery becomes powerless in comparison. Thus, a lot of new technology has introduced meta data to describe services semantically such as OWL-S,[1] SAWSDL.[2] They provide a more intelligent and efficient way for services matching.[3]

However, when a client (a user or an intelligent agent) intends to retrieve some service, he usually cannot get the services that he really expected. It is not a surprise because no one has the knowledge covering all domains so that he cannot express his requirement exactly in an unfamiliar area. Such uncertain and imprecise user requirements have to be confronted when discovering services, which has a negative impact on the quality of the result of retrieved services. Rough Set Theory proposed by Pawlak[4] is such a kind of tool to handle the uncertainty. In his perspective, *e*-service intelligence requires tools for approximate reasoning about vague concepts. The rough set based methods make it possible to deal with imperfect knowledge.[5]

Service discovery can be viewed as a process to find services whose capabilities are able to satisfy the requirement proposed by the client. The capabilities of a service may be regard as goals achieved after execution. Therefore, to discover services are equivalent to finding goals that clients really want to achieve. The goal-oriented requirements engineering (GORE) suggested by van Lamsweerde has long been recognized to be an essential component involved in the requirements engineering. Goals can be formulated at different levels of abstraction ranging from high-level, strategic concerns to low-level, technical concerns.[6] Goals at different levels can be seen as a set of requirement granules to be satisfied in the perspective at that level.

Granular computing[7,8] is a rising research area. The basic issues of granular computing often involve two related aspects, the construction of granules and computation with granules.[8] The former aspect deals with the formation, representation and interpretation of granule. The latter aspect deals with the utilization of granules in problem solving. In this paper, we introduce granular computing to enhance GORE so as to construct a more intelligent service platform.

2. REQUIREMENTS FORMALIZING WITH HIERARCHICAL GRANULES

Information granules can be treated as linked collections clumps of objects drawn together by the criteria of indiscernibility,

*Author to whom correspondence should be addressed.

 1936-6612/2012/7/135/005 doi:10.1166/asl.2012.2103

similarity or functionality.[7,9] In GORE, goals provide a precise criterion for sufficient completeness of a requirements specification.[6] Generally, a collection of goals with correlated functionality to achieve some objective can be regard as a granule in the requirement space. All granules in this hierarchical space provide views from different levels, which can zoom in the requirements specification from general to some specific points.

2.1. Hierarchical Goal Model for Requirement Refinement

GORE is an effective way for requirements elicitation, specifying, analysis, negotiation and modification. Goals capture the various objectives to achieve at different levels of abstraction. Once a set of goals at some level is obtained and validated, it can be decomposed into a set of subgoals with refinement links.[6] As shown in Figure 1, refinement links connect general goals to more specialized goal recursively, which forms the skeleton of the goal graph. Such multi-layer structure is derived from Artificial Intelligence (AI) planning where a goal is satisfied absolutely when all of its subgoals are satisfied. The goal structure can be viewed as generated by divide and conquer method to satisfy some complex requirement, where each goal in this structure is called an Action Granule (AG).[10]

In goal model, a goal can be formalized as a clause with a main verb and several parameters. Each parameter plays some role with respect to the verb.[11] For simplicity, we formalize the goal concept as a 2-tuple (*verb*, *target*). The element *target* designates entities affected by the goal. For example a goal for travel planning can be presented as: $(Plan_{verb}, Travel_{target})$. All of the goal concepts together with the refinement links in the hierarchy can be regarded as an ontology defining some requirement in a domain.

2.2. Partition Requirement in Goal Model

As discussed above, the goal model is a hierarchy in which a goal is satisfied when all its subgoals are satisfied. The leaves in goal model are called atomic goals (without subgoals). The goals in higher levels may be viewed as a successive bottom-up combination of leaf goals, and all of the goals in the hierarchy can be represented with atomic goals. If the height of a hierarchy is N, the root goal is top-down decomposed into a set of leave goals with $(N-1)$ times. All goals in goal model are granules of different grain size to cover requirement specification.

The granular space for a goal model is defined as:

$$G = \{X | X \textit{ is a node in the goal hierachy}\} \qquad (1)$$

Example 1: Given a goal model illustrated by Figure 2, all granules in the granular space are given:

$$G = \{Ga, Gb, Gd, G1, G2, G3, G4, G5, G6, G7\}$$

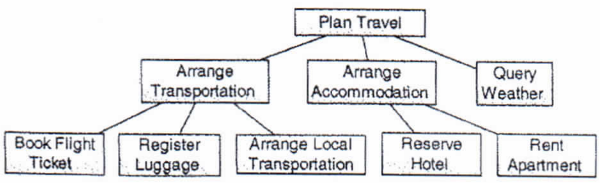

Fig. 1. A hierarchical goal model.

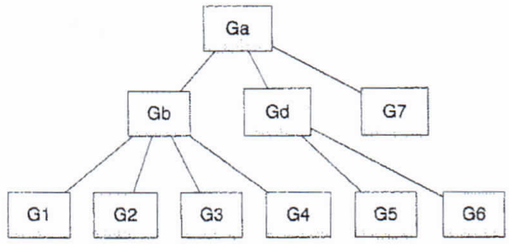

Fig. 2. A 3-layer goal model with ten nodes.

A non-leaf goal can be decomposed into a set consisting of its direct subgoals. The refinement for a granule into a set of finer granules may be viewed as zooming-in operation ω.[12]

Given a granule Gt, we denote zooming-in operation as:

$$\omega(Gt) = \{\textit{all direct subgranules of } Gt\} \qquad (2)$$

Further, if we denote all atomic goals $\{G1, G2, \dots, Gn\}$ (leaves in goal model) as $A(G)$, given a granule Gi, Gi, can be refined as a subset of $A(G)$ by finite zooming-in operations. The root granule is refined as the entire $A(G)$, and the leaves consist of only singleton subsets of $A(G)$. We denote the zooming-in operation refining a granule into the subset of $A(G)$ as $\omega_a(X)$, where $\omega_a(X) \subseteq A(G)$.

Example 2: Given the goal model as Example 1, refine granules Gb, Gd, Ga with the zooming-in operations ω, ω_a:

$$\omega(Gb) = \omega_a(Gb) = \{G1, G2, G3, G4\}$$
$$\omega(Gd) = \omega_a(Gd) = \{G5, G6\}$$
$$\omega(Ga) = \{Gb, G7, Gd\}$$
$$\omega_a(Ga) = \{G1, G2, G3, G4, G5, G6, G7\}$$

For any two granules $X, Y \in G$, we can define:

$$\omega_a(X) \subseteq \omega_a(Y) \Longleftrightarrow X \subseteq Y \qquad (3)$$

More general, if we denoted XS as a subset of G, then, for any $XS, YS \subseteq 2^G$, $\subseteq$ is defined:

$$\forall x \subseteq XS \rightarrow (\exists y \in YS)(x \subseteq y) \Longleftrightarrow XS \subseteq YS \qquad (4)$$

From the hierarchy, if we select an arbitrary non-leaf node g and all its direct and indirect sub nodes, then we can obtain a new granules set induced by g denoted as $U(g)$, obviously $U(g) \subseteq G$. Since g can be refined into $\omega_a(g)$, subsets of $U(g)$ can be selected to form a partitions of $\omega_a(g)$ with different granularities. The set of all partitions constructed from $U(g)$ is denoted as $P(g)$.

Any partition $\gamma \in P(g)$ has following properties:
(i) $\omega_a(X) \cap \omega_a(Y) = \varnothing \quad (X, Y \in \gamma, X \neq Y)$
(ii) $U_{X\in\gamma}\omega_a(X) = \omega_a(g)$

Example 3: Given the granules as Example 1 and set G_a as the root, construct all possible partitions $P(Ga)$:

$$\gamma 1 : \{Ga\} \overset{def}{=} \{\{G1, G2, G3, G4, G5, G6, G7\}\}$$
$$\gamma 2 : \{Ga, Gd, G7\} \overset{def}{=} \{\{G1, G2, G3, G4\}, \{G5, G6\}, \{G7\}\}$$
$$\gamma 3 : \{G1, G2, G3, G4, Gd, G7\}$$

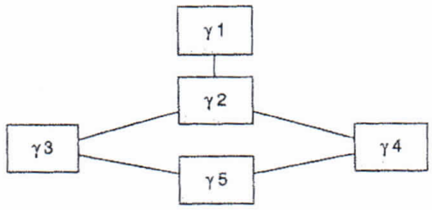

Fig. 3. The Hasse diagram of $P(Ga)$.

$$\stackrel{def}{=} \{\{G1\}, \{G2\}, \{G3\}, \{G4\}, \{G5, G6\}, \{G7\}\}$$

$$\gamma 4 : \{Gb, G5, G6, G7\}$$

$$\stackrel{def}{=} \{\{G1, G2, G3, G4\}, \{G5\}, \{G6\}, \{G7\}\}$$

$$\gamma 5 : \{\{G1\}, \{G2\}, \{G3\}, \{G4\}, \{G5\}, \{G6\}, \{G7\}\}$$

Note that all above partitions have the properties (i) and (ii).

Furthermore, $P(g)$ is a bounded lattice $L(g)$[13] whose order relation $\preccurlyeq$ is the inclusion relation given in definition (4). Figure 3 depicts the Hasse diagram of $P(Ga)$ for Example 3.

Different partitions in $P(g)$ can been viewed as different grain-sized solutions to achieve the goal g. Given two partitions $\gamma 1$, $\gamma 2$, if $\gamma 1 \preccurlyeq \gamma 2$, we say $\gamma 1$ is finer than $\gamma 2$, or $\gamma 2$ is coarser than $\gamma 1$. That is, $\gamma 1$ provides a more detailed view over requirements than $\gamma 2$. Some clients show no care for detail, their requirements can be modeled in a coarsely granular way, whereas others prefer involving into the details for certain purpose. Therefore, various granularities provide both coarse and detailed views to satisfy different users.

3. DELIVER SERVICES OVER IMPRECISE REQUIREMENTS

Since clients (users or intelligent agents) may not be domain experts, they often submit a very imprecise requirement about some domain. Even if the retrieved services fully match the submitted requirement, these services still cannot satisfy the goals that clients really expect to achieve. Hence, it is necessary to help clients clarify their requirement by adding goals they really desire to achieve and removing goals they do not want to involve.

3.1. Stratified Rough Set Approximation Space

In Pawlak's rough set model, the partition induced by an equivalence relation R on universe U is denoted as $U/R = \{C1, C2, \ldots, Cn\}$, where Ci is an equivalence class of R. Let an arbitrary subset X of U, every rough set is associated with two crisp sets, called lower and upper approximation.

The lower approximation of X:

$$\underline{apr}(X) = \{x \in Ci | Ci \subseteq X\} \tag{5}$$

The upper approximation of X:

$$\overline{apr}(X) = \{x \in Ci | Ci \cap X \neq \varnothing\} \tag{6}$$

The boundary region of X:

$$bnd(X) = \overline{apr}(X) - \underline{apr}(X) \tag{7}$$

Intuitively, the lower approximation of a set consists of all elements that definitely belong to the set, whereas the upper approximation constitutes of all elements that possibly belong to the set, and the boundary region of the set consists of all elements that cannot be classified uniquely to the set or its complement by employing available knowledge.[5]

Dubois and Prade[14] defined a rough set as a pair of subsets of U/R, and the pair of approximation is given:

$$\underline{apx}(X) = \{Ci | Ci \subseteq X\} \tag{8}$$

$$\overline{apr}(X) = \{Ci | Ci \cap X \neq \varnothing\} \tag{9}$$

The pair of approximations may be viewed as extensions of Pawlak's lower and upper approximations. In fact, they are consistent with each other[15] and every Ci can be seen as a granule.

As discussed in Section 2.2, the partitions in $P(G)$ form a lattice granulation structure $L(G)$. Given a subset X of G, we can take Eqs. (8, 9) to compute lower and upper approximations for each partition in $P(G)$. Further, the stratified rough set approximations can be produced from $L(G)$.

Example 4: Consider $P(G)$ given in Example 3 and the corresponding lattice $L(G)$ illustrated in Figure 3. Given the subset $X = \{G3, G5, G6\}$, the stratified rough set approximations are depicted in Figure 4.

3.2. Clarify User Requirements in Stratified Rough Set Approximations Space

The goal model is often constructed by domain experts to formalize domain requirements specification. Usually, a user requirement in some domain may be represented by a part of goals in a goal model. For example, given the goal model of domain "*Travel Planning*" as depicted in Figure 1, user requirement can be given as: $\{G1 : (Book,\ Flight\ Ticket), G2 : (Arrange,\ Accommodation)\}$

However, in fact, the goal model is often much more complex than this example. Since a client may have little knowledge of the domain that he is not familiar, he can only give a very imprecise requirement which does not cover all goals he desires to achieve. Therefore, we should recommend some goals which are most possibly expected by clients so that it may help to clarify requirements from clients. Here, we employ rough set model as an effective tool to handle such imprecision and uncertainty. Our proposal tries to refine user requirements in the stratified rough approximation spaces discussed previously.

The accuracy of rough set approximation[4] is defined as:

$$\alpha(X) = |\underline{apr}(X)| / |\overline{apr}(X)| \tag{10}$$

where $|\cdot|$ denotes the cardinality of a set. Obviously, $0 \leq \alpha(X) \leq 1$. If $\alpha(X) = 1$ then X is called definable

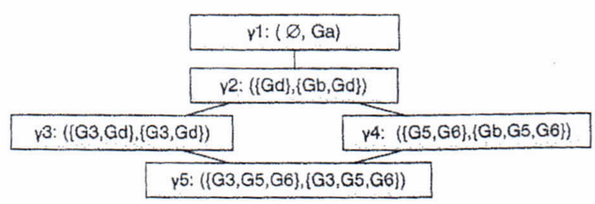

Fig. 4. The stratified rough set approximations of granules set $\{G3, G5, G6\}$.

$(\underline{apr}(X) = X = \overline{apr}(X))$, and otherwise, if $\alpha(X) < 1$ then X is indefinable or rough: $\underline{apr}(X) \subset X \subset \overline{apr}(X)$.

According to Eq. (1), if we denote G as the universe for all granules in the granular space, then rough set approximations can be used to measure the degree of X satisfying the requirements with a given goals set X. If we regard Ci in Eqs. (5–9) as a granule in G, $Ci \subseteq X$ represents that X can fully satisfy Ci, whereas $Ci \cap X \neq \varnothing$ represents that X can partially satisfy Ci.

The user requirement $\mathcal{R}$ can be clarified through four kinds of operations:

(1) refine $\mathcal{R}$ into finer granules;
(2) generalize $\mathcal{R}$ into coarser granules;
(3) add desired goals into $\mathcal{R}$;
(4) remove undesired goals from $\mathcal{R}$.

We demonstrate our method the with the goal model illustrated by Figure 2. If the initial user requirements are formalized as goals $\mathcal{R}$:$\{Gb, G5\}$, then we can obtain the stratified rough set approximations and corresponding accuracy as shown in Table I.

In Table I, the accuracy values indicate certainty of the user requirement in the stratified rough set approximations. The approximations with the accuracy value 1 are definable and they all have a pair of equal approximations. If we denote the approximation induced by partition γi as $apr_{\gamma i}$, where $apr_{\gamma i}(X) = \underline{apr}_{\gamma_i}(X) = \overline{apr}_{\gamma i}(X)$, then the set $apr_{\alpha=1}(X)$ is also a bounded lattice:

$$apr_{\alpha=1}(X) = \{apr_{\gamma i}(X) | \underline{apr}_{\gamma i}(X) = \overline{apr}_{\gamma i}(X)\} \quad (11)$$

and the order relation $\preccurlyeq$ is the inclusion defined in Eq. (4). That is, the least element in the lattice implies that the user requirement should be satisfied in the most detailed way. In contrast, the greatest element implies that the user requirement should involve the least details. A client can choose a proper grain-sized apr_{γ_i} from $apr_{\alpha=1}(X)$ to describe requirement. For example, according to Table I, if a client chooses $apr_{\gamma 5}$, it indicates that he is willing to involve more details comparing with $apr_{\gamma 4}$. Choosing different $apr_{\gamma i}$ can be viewed as clarifying user requirement through operation (1) or (2). When a client has selected a $apr_{\gamma i}$ from $apr_{\alpha=1}(X)$, the $apr_{\gamma i}$ is assigned as the new user requirement $\mathcal{R}'$ ($\mathcal{R}' = apr_{\gamma i}$). Moreover, a client may do the operation (4) to remove some goals that he does not desire from $\mathcal{R}$.

In fact, we may put more focus on the approximations having the accuracy value less than 1, because for each granule in boundary region contains a part of goals requested by clients in $\mathcal{R}$ and others not in $\mathcal{R}$. Since elements in a same set induced by rough set approximation are highly correlated, those granules in boundary region most probably contain potential requirements desired by clients but those clients may miss them in $\mathcal{R}$ due to lack of domain knowledge. Therefore, goals in these granules are best candidates for recommendation. These recommendations can be ranked by the accuracy value in a descendent order, because the higher is the accuracy, the less of modification is required to make user requirements more complete.

Example 5: If the user requirement $\mathcal{R}$:$\{G2, G3, G4, G5\}$ and the corresponding stratified rough set approximations are given in Table I. List the recommendations:

The recommended granules order by α are:

$$1.\{Gd\}_{\alpha=4/6} \quad 2.\{Gd\}_{\alpha=4/6} \quad 3.\{Gd\}_{\alpha=0}.$$

Removing duplicated granules and granules which have existed in $\mathcal{R}$, we can give the recommendations as below:

Rank	Recommendation	Comment
1	$G7$	Elements in Gd and remove $G5$ existed in $\mathcal{R}$
2	Gb, Gc, Gd	Elements in Ga

Clients adding desired goals from recommendations can be viewed as clarify $\mathcal{R}$ through operation (4). It should be noted that the added granules possibly absorb some small granules which are included in the new added ones. For example, if a client adds Gd into $\mathcal{R}$:$\{G2, G3, G4, G5\}$, $G5$ is absorbed by Gd because $G5$ is contained in Gd. The new $\mathcal{R}$ will be $\{G2, G3, G4, Gd\}$.

When clients do any operation described above, we get a more clarified user requirement $\mathcal{R}'$ from $\mathcal{R}$. If $\mathcal{R}'$ is still not clear enough, we can continue a new round clarity to compute the approximations, boundary and accuracy based on $\mathcal{R}'$ and repeat above 4 kinds operations. Such process makes user requirements to be clarified iteratively, and the process can stop at any time if clients think the requirement has become precise enough and no more refinement are required. We denote the final clarified user requirement as $\mathbb{R}$, and then we need to find services to satisfy $\mathbb{R}$.

4. AN ILLUSTRATIVE EXAMPLE

Let us consider the domain "*Travel Planning*." If the goal model of this domain is depicted in Figure 2, we can list all granules in the granular space:

$$G = \{Ga, Gb, Gc, Gd, G1, G2, G3, G4, G5\}, \text{where}$$
$$Ga \stackrel{def}{=} \{Plan, Travel\}$$
$$Gb \stackrel{def}{=} \{Arrange, Transporation\}$$
$$Gc \stackrel{def}{=} \{Arrange, Accommodation\}$$
$$G1 \stackrel{def}{=} \{Book, Flight\ Ticket\}$$
$$G2 \stackrel{def}{=} \{Register, Luggage\}$$
$$G3 \stackrel{def}{=} \{Arrange, Local\ Transporation\}$$
$$G4 \stackrel{def}{=} \{Reserve, Hotel\}$$
$$G5 \stackrel{def}{=} \{Rent, Apartment\}$$
$$G6 \stackrel{def}{=} \{Query, Weather\}$$

If a user submits a request: " I want to book a flight ticket and reserve a hotel room," by some NLP method,[16] it is formalized as the initial requirement, $\mathcal{R}$:$\{G1, G4\}$.

Table I. Stratified rough set approximations for partitions.

Partition	$\overline{apr}$	$\underline{apx}$	bnd	α
$\gamma 1$	$\{Ga\}$	$\varnothing$	$\{Ga\}$	0
$\gamma 2$	$\{Gb, Gd\}$	$\{Gb\}$	$\{Gd\}$	4/6
$\gamma 3$	$\{G1, G2, G3, G4, Gd\}$	$\{G1, G2, G3, G4\}$	$\{Gd\}$	4/6
$\gamma 4$	$\{Gb, G5,\}$	$\{Gb, G5\}$	$\varnothing$	1
$\gamma 5$	$\{G1, G2, G3, G4, G5\}$	$\{G1, G2, G3, G4, G5\}$	$\varnothing$	1

Construct all possible partitions $P(Ga)$ for the root Ga:

$$\gamma 1 : \{Ga\}$$
$$\gamma 2 : \{Gb, Gc, G6\}$$
$$\gamma 3 : \{G1, G2, G3, G4, Gc, G6\}$$
$$\gamma 4 : \{Gb, G4, G5, G6\}$$
$$\gamma 5 : \{G1, G2, G3, G4, G5, G6\}$$

The corresponding order relation of the bounded lattice $L(Ga)$ is: $\gamma 5 \subseteq \gamma 4, \gamma 3 \subseteq \gamma 2 \subseteq \gamma 1$.

Now we can easily compute out the stratified rough set approximations for the given requirement $\mathfrak{R}$ as follows:

Partition	$\overline{apr}$	$\underline{apx}$	bnd	α
$\gamma 1$	$\{Ga\}$	$\varnothing$	$\{Ga\}$	0
$\gamma 2$	$\{Gb, Gc\}$	$\varnothing$	$\{Ga\}$	1
$\gamma 3$	$\{G1, G4\}$	$\{G1, G4\}$	$\varnothing$	1
$\gamma 4$	$\{Gb, G4\}$	$\{G4\}$	$\{Gb\}$	1/4
$\gamma 5$	$\{G1, G4\}$	$\{G1, G4\}$	$\varnothing$	1

The recommended granules order by α are:

$$1.\ \{Gb\}_{\alpha=1/4} \quad 2.\ \{Gb, Gc\}_{\alpha=0} \quad 3.\ \{Ga\}_{\alpha=0}.$$

Add them into an ordered list: $\{1 : Gb, 2 : Gc, 3 : Ga\}$. The duplicated recommended granules should be removed when adding to the list, such as adding Gb in the second time. Then, the recommendations will be returned to the user:

Rank	Recommendation	Comment
1	$G2, G3$	Elements in Gb and remove Gl existed in $\mathfrak{R}$
2	$G5$	Elements in Gc and remove $G4$ existed in $\mathfrak{R}$
3	$Gb, Gc, G6$	Elements in Ga

From the recommendations, the user finds that he also desires to achieve following goals not included in $\mathfrak{R}$:

$$G2 \stackrel{def}{=} \{Register, Luggage\}$$
$$G3 \stackrel{def}{=} \{Arrange, Local\ Transporation\}$$
$$G6 \stackrel{def}{=} \{Query, Weather\}$$

Besides, the user prefers renting an apartment to reserving a hotel room. So he removes $G4$ and adds $G5$. The refined requirement becomes: $\mathfrak{R}$:$\{G1, G2, G3, G5, G6\}$.

Using this new $\mathfrak{R}$, the user continues to clarify his requirement in second round:

partition	$\overline{apr}$	$\underline{apx}$	bnd	α
$\gamma 1$	$\{Ga\}$	$\varnothing$	$\{Ga\}$	0
$\gamma 2$	$\{Gb, Gc, G6\}$	$\{Gb, G6\}$	$\{Gc\}$	4/6
$\gamma 3$	$\{G1, G2, G3, Gc, G6\}$	$\{G1, G2, G3, G6\}$	$\{Gc\}$	4/6
$\gamma 4$	$\{Gb, G5, G6\}$	$\{Gb, G5, G6\}$	$\{Gb\}$	1
$\gamma 5$	$\{G1, G2, G3, G5, G5\}$	$\{G1, G2, G3, G5, G5\}$	$\varnothing$	1

In this round, the user no longer wants to do any change. Two definable approximations are obtained, which represent the clarified user requirement in different granularities: $\{Gb, G5, G6\}$ and $\{G1, G2, G3, G5, G6\}$. If the user prefers that a service provider can arrange transportation for him as a whole, then $\mathbb{R} \stackrel{def}{=} \{Gb, G5, G6\}$ should be chosen as the final clarified requirement.

5. CONCLUSION

In this paper we try to help clients to clarify their initial imprecise requirement in a hierarchical granular space over goal model. Each goal in the goal model corresponds to a granule in the space, and these granules cover requirements in various grain sizes. With such hierarchical granulation structure, granular computing and rough set theory are employed to handle uncertainties. It can distinguish what has been definable and what is still uncertain. The clarification process can produce a clarified requirement of a desired granularity by iteratively executing four kinds of operations in the stratified rough set approximation space. In fact, we also have designed a matching algorithm based on six types of granule approximation as service matching strategies.[17]

Acknowledgments: The paper is partially supported by China National Science Foundation (Granted Number: 60873230, 61073021), Scientific Research Projects of STCSM (Granted Number 08JC1411700, 10511501503 and 10692101400), National High-Tech Research and Development Plan of China (Granted Number: 2006AA01A124) and Program for New Century Excellent Talents in University (Granted Number: NCET-08-0347).

References and Notes

1. D. Martin, M. Burstein, D. McDermott, D. McGuinness, S. Mcllraith, M. Paolucci, E. Sirin, N. Srinivasan, and K. Sycara, OWL-S 1.2 Release. In http://www. daml.org/services/owl-s/1.2 **(2008)**.
2. J. Farrell and H. Lausen, Semantic annotations for WSDL and XML schema. In http://www.w3.org/TR/sawsdl/ **(2007)**.
3. M. Paolucci, T. Kawamura, T. R. Payne, and K. Sycara, *Lect. Notes Comput. Sc.* 2342, 333 **(2002)**.
4. Z. Pawlak, Rough sets: Theoretical aspects of reasoning about data, Springer **(1991)**.
5. Z. Pawlak and A. Skowron, *E-Service Intelligence* 35 **(2007)**.
6. A. van Lamsweerde, *Fifth IEEE International Symposium on Requirements Engineering, Proceedings* 249 **(2000)**.
7. L. Zadeh, *Fuzzy sets and systems* 90, 111 **(1997)**.
8. Y. Yao, Granular computing: basic issues and possible solutions. *Proceedings of the 5th Joint Conference on Information Sciences*, Citeseer **(2000)**, Vol. 1, pp. 186.
9. A. Skowron and J. Stepaniuk, *Int. J. Intell. Syst.* 16, 57 **(2001)**.
10. L. Zadeh, *Intelligent Systems and Soft Computing* 3 **(2000)**.
11. C. Rolland, C. Souveyet, and C. Ben Achour, *IEEE Transactions on Software Engineering* 24, 1055 **(1998)**.
12. Y. Yao, *Transactions on Rough Sets I* 232 **(2004)**.
13. Y. Y. Yao, *Int. J. Intell. Syst.* 16, 87 **(2001)**.
14. D. Dubois and H. Prade, *International Journal of general systems* 17, 191 **(1990)**.
15. Y. Yao, Combination of rough and fuzzy sets based on a-level sets, Rough sets and data mining: analysis for imprecise data, Kluwer Academic Publishers **(1997)**.
16. L. Hu, J. Cao, and Z. Gu, Modeling semantic web service using semantic templates. *Fourth International Conference on Semantics, Knowledge and Grid* **(2008)**.
17. L. Hu, J. Cao, and Z. Gu, Service Discovery and Recommendation in Rough Hierarchical Granular Computing, *Services Computing Conference (APSCC)* **(2010)**.

Received: 28 February 2011. Accepted: 19 June 2011.