UDC 681.03

V.A. Shekhovtsov, N.A. Bazhenov
National Technical University “Kharkov Polytechnic Institute”
shekvl@yahoo.com, nabazhenov@gmail.com

Using NLP to Define the Scope for Stakeholder Assessment of Simulated Service Qualities

The paper is devoted to defining the scope of research activities aimed at involving business stakeholders in a software process in a form of assessing the perceived quality of the service-oriented system in its usage context when the initial specification of the system is available in natural language form. We propose to use NLP techniques to extract the scope from this specification and to represent it in the format of specific predesign models compatible with the rest of the simulation solution.

Introduction

Gathering the opinions of business stakeholders (not necessary possessing any IT background) is considered an important part of the software process by both software engineering researchers and practitioners. The main subject of such opinions is currently the prospective system’s functionality; it forms the foundations for widely studied functional requirements elicitation problem. Less studied but still important kind of opinions are assessments of the quality of the prospective system. The reason of this importance is that if stakeholders cannot express such opinions early in the software development lifecycle these they could be easily lost. As a result of this loss, the understanding of the desired quality of the system becomes biased towards the view of the IT people. This could lead to the stakeholder dissatisfaction with the quality of the *software under development* (SUD) late in the development lifecycle. This is dangerous, as the problems with satisfying the customer revealed that late could easily lead to the complete failure of the whole project.

To address this problem, we proposed ISAREAD-S framework (*Interactive Simulation-Aided Requirements Engineering and Architectural Design for Services*) [1], [2]. It is aimed at investigating ways of supporting stakeholder involvement in the software process by allowing business stakeholders to assess the perceived qualities of the prospective system in its usage context. To implement such support we plan to elaborate a set of simulation-based methods aimed at making QoS (quality of service) assessment procedures accessible to the business stakeholders and using their assessments as a driving force for activities related to requirements engineering and architectural design.

The research goal is to define *a scope* for these simulation activities (*ISAREAD-S scope*), i.e. the initial set of services and qualities of interest together with possible usage contexts. We propose to obtain this scope by analyzing the existing description of the problem domain by means of *Natural Language Processing* (NLP) techniques.

The structure of the paper is as follows. Section 1 shows the principles of the existing procedures (mechanisms) for organizing the interaction with stakeholders; these mechanisms form the target activities for the scope definition solutions proposed in the paper, Section 2 formulates the problem statement, Section 3 describes the models used to define the ISAREAD-S scope, Section 4 outlines the proposed NLP-based approach for transferring the specification for the SUD or the problem domain into the ISAREAD-S scope definition, and finally section with necessary conclusions.

1. Interactive assessment of simulated service qualities

ISAREAD-S approach includes three kinds of mechanisms.

Service-level mechanisms organize assessments for the simulated qualities of the particular services. We define two mechanisms of this kind: model composition and model execution mechanism. Model composition mechanism combines parameterised simulation models with corresponding interaction models for the particular qualities to form interactive simulation models allowing business stakeholders to participate. The participation depends on the quality being processed; it involves experiencing the quality and making an assessment using specific scale. Model execution mechanism is based on model driven paradigm as it treats interactive simulation model as executable model; it is responsible for interactive execution of such models to organize assessments of the qualities of interest.

Process-level mechanisms put these service-level assessments into the context of simulated usage processes. We propose to represent such usage processes using Business Process Modeling (BPM) notations such as BPMN or Petri Nets. Similarly to the service-level case, process-level model composition and model execution mechanisms are defined. The first mechanism embeds service-level interactive simulation models for the services of interest into the business process model to form interactive simulation model for service usage process. Model execution mechanism executes this model to establish a user session. In the course of this session, the user is asked to perform assessments of the qualities of interest for the services of interest.

An iterator mechanism controls the interaction sessions and gathers all necessary assessments for the particular SUD (treated as a set of services of interest).

On top of these mechanisms the policies for requirements elicitation, negotiation, and validation, architecture assessment and lifecycle support are defined. In addition, the adapters are responsible for the transformation of the external information related to the project (such as the amount of available resources) into the numerical values for the factors influencing the simulations.

2. The problem of ISAREAD-S scope definition

The above description of the ISAREAD-S framework misses one important issue: it is not clear how to define the set of services and qualities of interest and possible usage context (which form the scope of the future simulation-related activities) prior to the assessment. It was implicitly assumed that this set is known beforehand to software engineers. Unfortunately, it is not always the case; as a result, defining such set by hand every time the new project begins could be cost-ineffective.

The ways of handling this problem vary significantly depending on availability of the description of the prospective system (or at least the problem domain). In this work, we restrict ourselves with the case when such description is available in natural language form, which leads to the following research question which determines **the problem statement**: *How to determine the set of services and qualities of interest together with possible usage contexts from the existing natural language specification of the prospective SUD and make it available to the ISAREAD-S framework?*

To address this question, we propose to use Natural Language Processing techniques to process the specification documents with a goal of obtaining structural representation of the scope of the ISAREAD-S application (Fig. 1). This way, such process could be automated which reduces the costs of applying the framework.

3. Representing ISAREAD-S scope with predesign models

Before investigating possible ways to process the description of the prospective system or the problem domain we need to define the target of this processing, namely, the format we are planning to use to represent the scope of the ISAREAD-S framework to be used by its components.

Figure 1 – Problem definition

As our framework focuses on supporting stakeholder involvement into the software process activities we should implement the interface which, on the one hand, is suitable for validation by business stakeholders (i.e. the people without special IT-skills) and, on the other hand, is suitable for implementing the support for the simulation of service qualities. We use special requirements models for this purpose. These models propose to the user a tabular form of communication, which is understandable for domain experts.

To implement user-centered view of the SUD functionality (*functional ISAREAD-S scope*), i.e. the set of services of interest, we use Klagenfurt Conceptual Predesign approach developed in Klagenfurt University [3], [4] which defines a metamodel (Fig. 2) for *Klagenfurt Conceptual Predesign Model (KCPM)*. This model consists of static and dynamic parts, which include process-invariant and service process information respectively. *Thing-type* and *connection-type* form the static part. Thing-type generalizes the class notion as well as attribute and value notions. The examples of this concept are persons, objects, resources, attributes, characteristics, abstracts and other entities. Connection-types represent relations between thing-types. *Operation-type* and *cooperation-type* constitute the dynamic part. Operation-types are intended to define service operations, their actors and parameters, which are expressed by thing-types. Cooperation-types are used to model business processes which orchestrate services. A cooperation-type consists of a triple of sets <Prc, {A,O}, Poc>, where Prc – the set of pre conditions, Poc – the set of post conditions, {A,O} – the set of pairs consisting of an operation-type and an actor executing this operation.

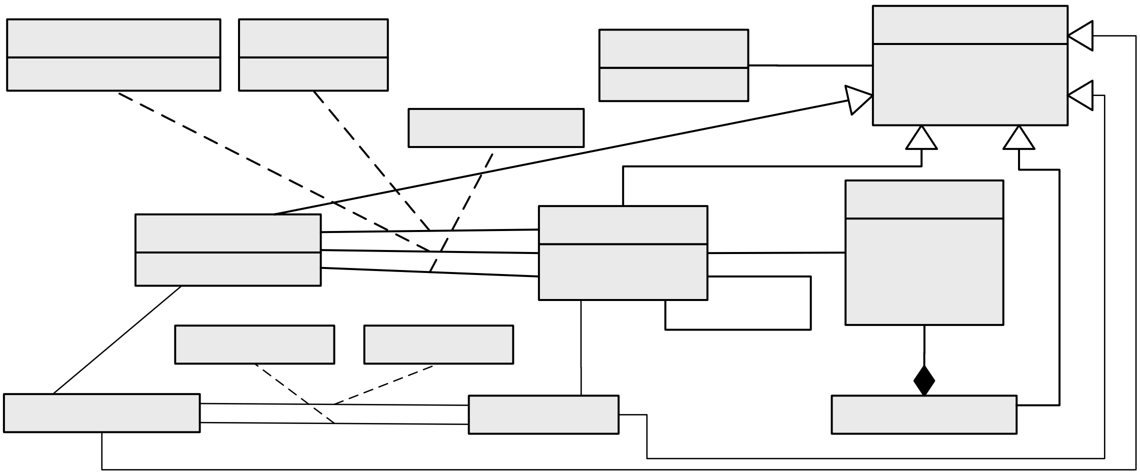

Figure 2 – KCPM metamodel for a functional ISAREAD-S scope

To implement user-centered view at the qualities of the prospective SUD (*non-functional (quality-related) ISAREAD-S scope*), i.e. the set of qualities of interest, we use *Quality-Aware Predesign Model for Services (QAPM-S)* [5]. It is connected to KCPM (functional model) via common abstract concept ModelingElement and dynamic elements – operation-type and cooperation type (Fig. 3). This model is based on the notions defined in ISO/IEC 9126 standard such as quality category, quality characteristic, quality metric, etc.

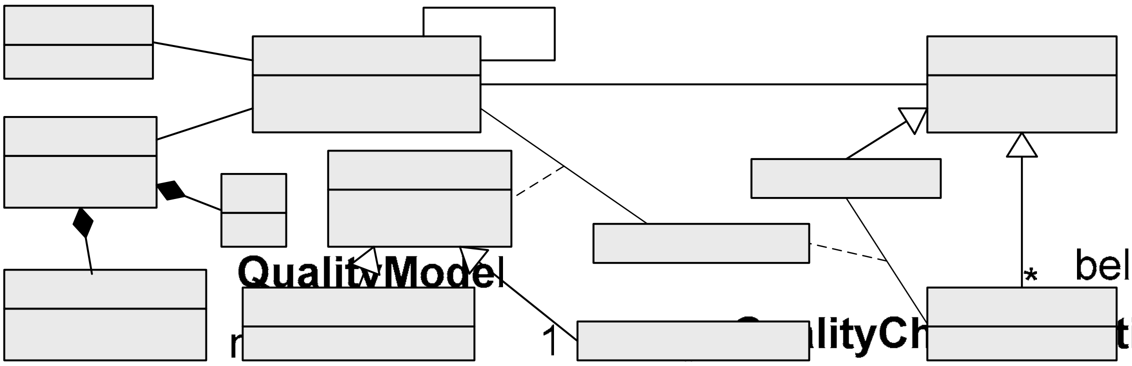

Figure 3 – QAPM-S metamodel for a quality-related ISAREAD-S scope

4. Using NLP techniques for ISAREAD-S scope definition

Morphosyntactic and semantic analysis. To obtain structural representation of ISAREAD-S scope we need to bridge the gap between natural language knowledge data generated by stakeholders and structured modeling data used by designers and developers. These acts are being solved using different NLP methods [6]. For the moment we use one of the approaches based on context-free grammars and Chomsky generative syntax. More detail information is described in previous paper [7]. This approach combines probabilistic part-of-speech tagging with sophisticated rule-based chunking and produces from free requirements text structured ontology-oriented tree-based output, available in XML format [4]. Its main benefits are:

− enriching classical part-of-speech (POS) tagging with additional lexical information;

− compound nouns identification related to term identification;

− two-level verb subclass identification (firstly taking into account the number of noun pseudo-objects (nPO); secondly after chunking procedure correction according to the verb arguments number);

− post-modifier of nouns;

− flexible settings allowing to manipulate with different lexical features (phrasal nodes, etc.);

− verb roles disambiguation.

But in spite of these advantages there are some difficulties with the processing of complex language constructions and English language dependency. To process other languages the developed engine requires the appropriate language resources both the language data itself and its structure rules information. For increasing accuracy of text processing and targeting an extended set of human languages more flexible and semantic-oriented approaches (like MTT [8], or UNL [9]) can be embedded in ISAREAD-S framework.

The example of this requirements elicitation step is listed below:


```
- <sentence>
    - <sentence type="subordinate">
        <con0 derivedPOS="n0" type="subord">If</con0>
      - <n3>
          <det0 form="general" type="def">the</det0>
          <n0 base-form="order" derivedPOS="v0" num="sg" type="common">order</n0>
        </n3>
        <v0 base-form="come" form="ind" num="sg" phrasalverb="come in" ps="3"
      temp="pres" verbclass="iV">comes</v0>
        <p0 derivedPOS="pt0" phrasalverb="come in">in</p0>
    </sentence>
```

```
- <n3>
    <det0 form="general" type="def">the</det0>
  - <n0 desc="compound" type="common">
      <n0 base-form="bookkeeping" corelex="act" derivedPOS="a0" num="sg"
    type="common">bookkeeping</n0>
      <n0 base-form="department" corelex="grs" num="sg"
    type="common">department</n0>
    </n0>
  </n3>
  <v0 base-form="check" derivedPOS="n0" form="ind" temp="pres" num="sg" ps="3"
verbclass="tvag2">checks</v0>
- <n3>
    <det0 form="general" type="def">the</det0>
    <n0 base-form="payment" corelex="poa" num="sg" type="common">payment</n0>
  </n3>
</sentence>
```


The requirements sentence *"If the order comes in, the bookkeeping department checks the payment"* was transformed into structured tree-based head-lexicalized format. The second part of sentence is governed by verb "check" which has two arguments: subject – noun phrase "the bookkeeping department" and object – noun phrase "the payment". This XML material is ready for interpretation and modeling notions extraction.

Interpretation. After free text processing the obtaining output is ready to be interpreted into different modeling concepts and parameters. For modeling context-aware functional service aspects the user-centered approach is used [3]. As this approach's pivot is the stakeholder involvement in development process, its main view is the set of glossaries (organized into QAPM-S model [5]), which are filled with modeling concept from processed text. For this purpose we use various interpretation rules, mainly based on predicate argument structure of verbs and its agentivity. The list of verb classes identified during the linguistic processing and used for interpretation can be found in [7]. The head of noun phrases and the parts of compound nouns are transferred into *thing-type* glossary. The containing and attributive relations create entries in *connection-type* glossary. The transitivity of verbs flags to what element of behavior model part it belongs to. For example, verbs with categories "tvag2" (monotransitive verb with agent subject), "tv3" (ditransitive verb), "iV" (intransitive verb) [1] denote an *operation-type*.

The set of such operation types correspond to the set of services of interest comprising the ISAREAD-S scope. The noun on subject place is mapped to an *executing actor*, whereas on object place to a *calling actor* or *parameter* depending on context and other trigger rules. The verb without agent subject, e.g. "tv2", denotes the *condition* in the *cooperation-type* glossary [10].

Quality characteristics being modeled are structured into the QAPM-S model and can also be extracted from the NLP output. The quantor phase contains the quantitative quality characteristics in this model. The service usage context represented by operation-type and cooperation type used from its original glossaries is also interpreted using sentence and phrase relationships.

The main rules of interpretation are listed in tables 1 – 2. The example of interpretation engine applying for the mapping from tree-based structured requirements into glossaries' entries is represented in fig. 4. The engine process the requirements sentence applying step by step the interpretation rules climbing up the rule order.

Consider more detail the list of rules used to form functional requirements model. *Rule 1* reflects the basic principle that every noun is thing or abstract notion and therefore mapped into thing-type entry. The attribute assigned during noun type identification can be transformed to “classification” field in thing type. *Rule 2* means that in the case of compound, i.e. congruence between parts of compound concerning proper or common type. *Rule 3* denotes the relation between the head and dependent member in compound noun structure. *Rule 4* identifies the post-modifier in the case of prepositional phrase (p2) [11]. “p2” can play role both as the argument of verbal phrase and as adjunctive or adverbial modifier of other sentence argument. This ambiguity is tried to solve by developed approach [6]. And in the case of modifier it should reproduce the entry in connection type glossary. *Rule 5* captures the abstraction generalization and component/object which are treated as connection types [12]. These cases occur if the verb “to be” in 3rd person functions as the main verb in the sentence. *Rule 6* is intended for possession relation between subject and object. It occurs then the verb “to have” functions as the main verb. Other default cases for the verb with more than one argument are applied by *Rule 7*. These cases present the relation between verb arguments with respect to its valency. *Rule 8* provides the operation-type identification. The noun phrase, which is agent subject, becomes an “actor” whereas other noun phrases, which have other argument roles, are mapped into parameters of operation-type. *Rule 9* is executed in the case of past participle of transitive and intransitive verbs and denotes the pre-condition in cooperation-type glossary. *Rule 10* specifies the condition for operation-type if the verb is not an agentive verb and all the arguments of the verb become candidates for involved thing-types (according to the procedure described in [11]). *Rule 11* indicates the conditions of property/state type; it means then adjective or adverbial phrase describe the properties of thing-type. *Rule 12* decodes the event type of condition; it means then act is carried out, but the subject does not act. *Rule 13* allows us to suppose the passive voice construction as the candidates for conditions of cooperation-types. *Rule 14* emphasizes the if/when constructions as the basic valid sentence patters of cooperation-type.

Consider the list of rules for quality requirements. *Rule 1* shows how the quantitative value of quality characteristic can be handled. We presuppose that quantor modifier of noun phrase can serve such value. The subject in that sentence becomes the QualityMetric element. *Rule 2* specifies the qualitative parameters of quality requirement. This part of investigations is still in progress and requires the additional intelligence methods to distinguish elements of QAPM-S among the whole variety of modeling concepts, such as applying specified ontology patterns, which are able to detect the traditional quality characteristics (e.g. defined in ISO standards: availability, accessibility, performance, etc.), or word class identification via WordNet-based systems, which could provide the assignment of terms to needed cluster.

Table 1 – Part of interpretation rules for functional requirements model

№	Rule	Description	Glossary	Example
1.	n0 → thing type	n0.corelex → thing type.classification	Thing type	Order
2.	n0(desc=compound) → thing type	Compound noun	Thing type	Order processing service
3.	n0(desc=compound) → attribute	Attribute	Connection type	Order item
4.	n3, n2.child=p2 → post-modifier	Attribute	Connection type	The man with the hat
5.	v0(verbclass=copV) → generalization/aggregation	Abstract generalization “is_a” or component “is_part_of”	Connection type	A truck is a car

Table 1 (Cont.)

<table>
<tr><th>№</th><th>Rule</th><th>Description</th><th>Glossary</th><th>Example</th></tr>
<tr><td>6.</td><td>v0(verbclass=possV) → possession</td><td>Possession</td><td>Connection type</td><td>Each hard drive has a capacity</td></tr>
<tr><td>7.</td><td>v0(verbclass=psychV | tvag2 | locV | tv3 | sentV | tv2) → argument relation</td><td>Relation between verb arguments with respect to its valency</td><td>Connection type</td><td>The order department relates the item to the order</td></tr>
<tr><td>8.</td><td>v0(verbclass=iV | tvag2 | tv3 | sentV | psychV) → operation type; n3$_{subject}$ →actor, n3$_{object}$ | p2 | sentence(type=inf) →parameters</td><td>Activity/Action is executed by agent-subject.</td><td>Operation type</td><td>The order department for each ordered item checks its availability on stock</td></tr>
<tr><td>9.</td><td>v0(verbclass=iV | tvag2 | tv3 | sentV | psychV, temp=perf) → condition; n3 | p2 | sentence(type=inf) →involvedTypes</td><td>Completion of activity</td><td>Cooperation type</td><td>If department has checked the order,…</td></tr>
<tr><td>10.</td><td>v0(verbclass=tv2) → condition, n3 →involvedTypes</td><td>Post or precondition</td><td>Cooperation type</td><td>Payment is needed</td></tr>
<tr><td>11.</td><td>v0(verbclass=locV | possV | copV) → condition</td><td>Property/state</td><td>Cooperation type</td><td>All articles of order are in stock</td></tr>
<tr><td>12.</td><td>v0(verbclass=eV) → condition</td><td>Event</td><td>Cooperation type</td><td>The window broke</td></tr>
<tr><td>13.</td><td>v0(mode=pass) → condition</td><td>Passive construction</td><td>Cooperation typt</td><td>The article is ordered</td></tr>
<tr><td>14.</td><td><con0> <n3> <v0> [<n3> | <p2> | <sentence>], <adv2> <n3> <v0> [<n3> | <p2> | <sentence>]</td><td>If – then construction</td><td>Cooperation type</td><td>If each ordered item is on stock, then order department relates that item to the order.</td></tr>
</table>

Table 2 – Part of interpretation rules for quality requirements model

<table>
<tr><th>№</th><th>Rule</th><th>Description</th><th>Element</th><th>Example</th></tr>
<tr><td>1.</td><td><n3$_{subject}$> <v0(verbclass=copV)> <n3$_{object}$>, n3$_{object}$.child=q2 → n3$_{subject}$ -QualityMetric, QualityInContext.value=q2</td><td>The value of quality parameter</td><td>Quality Metric</td><td>The response time to the stock items replenishment is below 2 seconds.</td></tr>
<tr><td>2.</td><td><n3$_{subject}$> <v0.verbclass=copV> a2 → n3$_{subject}$ -QualityMetric, QualityInContext.description=a2</td><td>The characteristic of quality parameter</td><td>Quality Metric</td><td>Accessibility is highest possible</td></tr>
</table>

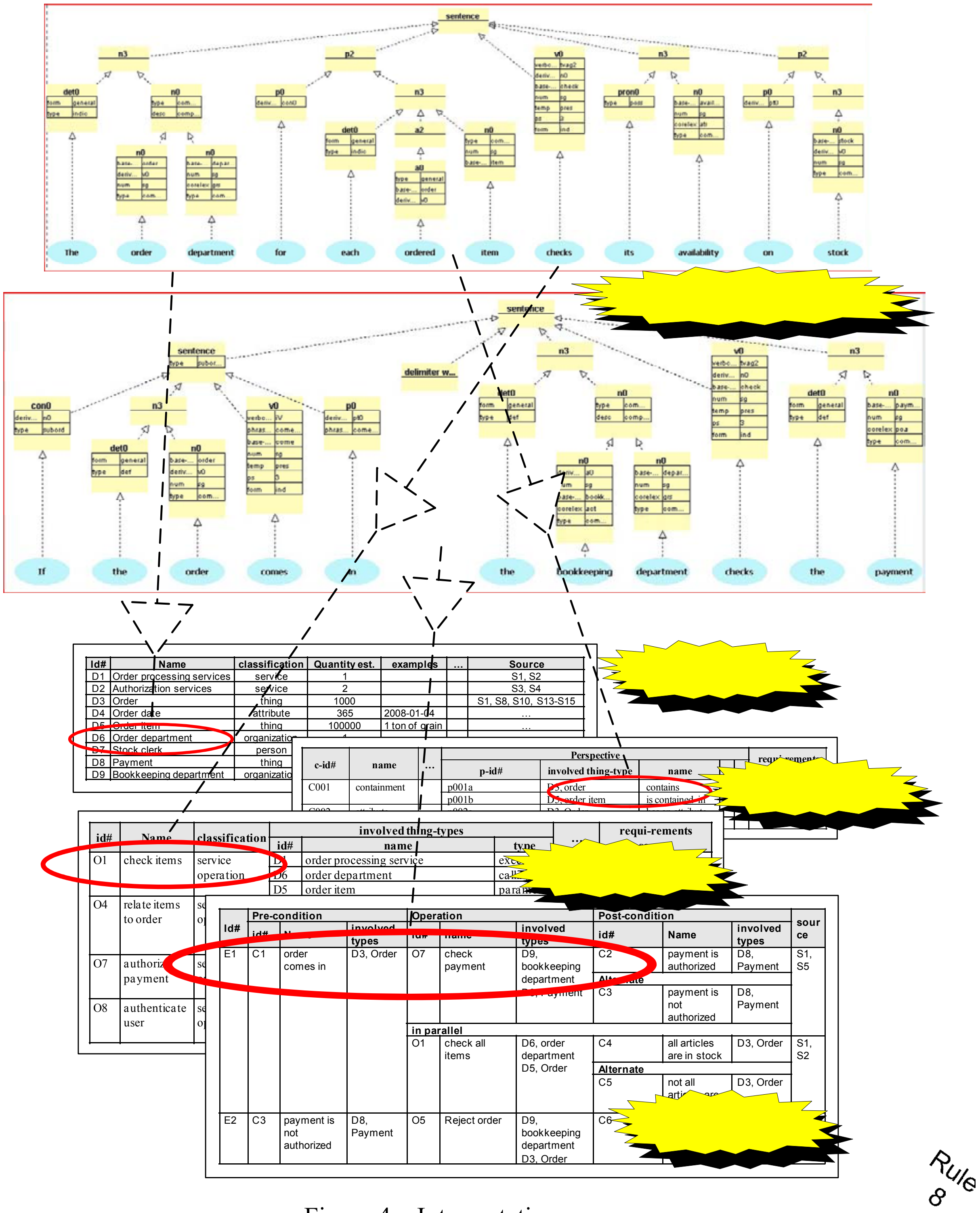

Figure 4 – Interpretation process

Conclusions

As a result of applying the proposed technique, QAPM-S representation of the set of services and qualities of interest (and, optionally, some of their usage contexts) could be obtained. This representation serves as a scope for subsequent quality simulation and assessment activities. Automating the task of defining the ISAREAD-S scope reduces the up-front costs for applying ISAREAD-S framework. Reducing these costs can be considered an important step in the direction of increasing the feasibility of its deployment.

Literature

1. Using Linguistic Knowledge for Fine-tuning Ontologies in the Context of Requirements Engineering / J. Vöhringer, D. Gälle, G. Fliedl, C. Kop, M. Bazhenov // International Journal of Computational Linguistics and Applications. Bahri Publications. – 2010. – Vol. 1. – P. 249-267.
2. Towards Simulation-Based Quality Requirements Elicitation: A Position Paper / R. Kaschek, C. Kop, V.A. Shekhovtsov, H.C. Mayr // REFSQ 2008. – LNCS, Vol. 5025. – Springer. – 2008. – P. 135-140.
3. Kop C. Mapping Functional Requirements: From Natural Language to Conceptual Schemata / C. Kop, H.C. Mayr // Proc. SEA'02. – 2002. – P. 82-87.
4. Mayr H.C. Conceptual Predesign – Bridging the Gap between Requirements and Conceptual Design / H.C. Mayr, C. Kop // Proc. ICRE '98. – IEEE CS Press, 1998. – P. 90-100.
5. Shekhovtsov V.A. Capturing the semantics of quality requirements into an intermediate predesign model / V.A. Shekhovtsov, C. Kop, H.C. Mayr // Proc. SIGSAND-EUROPE'2008 Symposium, Lecture Notes in Informatics (LNI) P-129. – Bonn : GI-Edition, 2008. – P. 25-37.
6. Shekhovtsov V.A. Interactive assessment of simulated service qualities by business stakeholders: principles and research issues / V.A. Shekhovtsov // Проблеми програмування. – 2010. – № 2-3. – С. 288-298.
7. Bazhenov N.A. Combining probabilistic tagging with rule-based multilevel chunking for requirements elicitation / N.A. Bazhenov // Искусственный интеллект. – 2010. – № 2. – С. 6-14.
8. Мельчук И. Опыт теории лингвистических моделей Смысл-Текст / Мельчук И. – М. : Наука,1974.
9. UNDL Foundation [Электронный ресурс]. – Режим доступа : http://undl.org.
10. Extended Tagging and Interpretation Tools for Mapping Requirements Texts to Conceptual (Predesign) Models / G. Fliedl, C. Kop, H.C. Mayr [et al.] // Proc. of 10th Int. Conf. on Applications of Natural Language to Information Systems NLDB 2005. – Springer, Heidelberg Lecture Notes in Computer Science, 2005. – Vol. 3515. – P. 173-180.
11. Deriving static and dynamic concepts from software requirements using sophisticated tagging / G. Fliedl, C. Kop, H.C. Mayr [et al.] // Data & Knowledge Engineering, 2007. – Vol. 61. – P. 433-448.
12. Linguistically based requirements engineering – The NIBA-project / G. Fliedl, C. Kop, H.C. Mayr [et al.] // Data & Knowledge Engineering, Special issue on NLDB '99: applications of natural language to information systems, 2000. – Vol. 35. – P. 111-120.

В.А. Шеховцов, М.О. Баженов

Використання засобів обробки природної мови для визначення області застосування користувацького оцінювання змодельованої якості обслуговування

Стаття присвячена визначенню області проведення досліджень, пов’язаних з підключенням зацікавлених осіб до процесу розробки програмного забезпечення через оцінювання сприйманої якості сервіс-орієнтованих систем в контексті їхнього використання, коли початкова специфікація системи задана природною мовою. Пропонується використання технології аналізу природної мови для отримання інформації про область застосування з цієї специфікації у форматі спеціальних моделей предпроектування, які є сумісними з основними модулями імітаційного рішення.

В.А. Шеховцов, Н.А. Баженов

Использование средств обработки естественного языка для определения области применения пользовательского оценивания смоделированного качества обслуживания

Статья посвящена определению области проведения исследований, связанных с подключением заинтересованных лиц к процессу разработки программного обеспечения через оценивание воспринимаемого качества сервис-ориентированных систем в контексте их использования, в случае, если начальная спецификация системы задана естественным языком. Предлагается использование технологии анализа естественного языка для получения информации об области применения из этой спецификации в формате специальных моделей предпроектирования, которые являются совместимыми с основными модулями имитационного решения.

Статья поступила в редакцию 02.07.2010.