TRS: System for Recommending Semantic Web Service Composition Approaches

Sandeep Kumar, and R. B. Mishra

Abstract—A large number of semantic web service composition approaches are developed by the research community and one is more efficient than the other one depending on the particular situation of use. So a close look at the requirements of ones particular situation is necessary to find a suitable approach to use. In this paper, we present a Technique Recommendation System (TRS) which using a classification of state-of-art semantic web service composition approaches, can provide the user of the system with the recommendations regarding the use of service composition approach based on some parameters regarding situation of use. TRS has modular architecture and uses the production-rules for knowledge representation.

I. INTRODUCTION

SEMANTIC web service (SWS) composition process is generally performed when no available single or composite service can satisfy the required request and a combination which can satisfy request can be generated from available single or composite services.

A large number of different composition techniques have been developed by the researchers and the choice of proper one to use highly affect the success of composition process. So, a system, providing recommendations regarding this, can be highly useful to the research community. [1], [2], [3], [4], [5] have performed the work on classifying different SWS composition approaches, which may be some helpful in choosing a particular composition approach. But each of these works only covers some of the particular category of composition approaches and also not provides any system framework helpful to the user in performing the selection of best composition approach to use. In this paper, we have presented a Technique Recommendation System (TRS), which can provides recommendations to the developers involved in the semantic web development activities regarding the service composition approach to use under a particular situation. The work will consider theoretically all classes of

Manuscript received April 30, 2008

Sandeep Kumar is with Department of Computer Engineering, Institute of Technology, Banaras Hindu University (IT-BHU), Varanasi-221005, India (phone: +91-9415643323; e-mail: sandeepkumargarg@gmail.com, sandeep.garg.cse06@itbhu.ac.in).

R. B. Mishra is with Department of Computer Engineering, Institute of Technology, Banaras Hindu University (IT-BHU), Varanasi-221005, India (e-mail: ravibm@bhu.ac.in).

composition techniques during providing recommendations. TRS has modular architecture and uses the production rules for knowledge representation. In support of work, a prototype system has also been implemented.

Apart from introduction in part I, the paper has been structured as follow. Part II provides a detailed description with architecture of TRS and a classification hierarchy of SWS composition approaches to be used in the TRS. Part III deals with the implementation issues involved in the TRS and also present a comparison and evaluation against the existing work. The work has been concluded in the part IV with some discussion on the future work.

II. TECHNIQUE RECOMMENDATION SYSTEM

An appropriate selection of a SWS composition approach can save a lot of cost, time, and developer-effort, and can produce high quality results. A recommendation system, which can provide recommendations on the suitability of the composition approach under given situations, can be of good use in this regard. This section introduces a rule-based recommendation system, Technique Recommendation System, which asks some of the questions to the user about the service composition to be performed and using answers provide recommendation and advice on the suitability of any composition approach. The system can provide recommendations about the composition approach with decreasing levels of abstraction based upon the levels in the classification hierarchy as described in following section. As the levels of hierarchy increases, the level of abstraction decreases and the more details of the process come into picture.

A. A Classification Approach

In this section, we have presented a classification system (TCS) in the form of taxonomy for SWS composition approaches. Taxonomical Classification System (TCS) provides the classification of large range of SWS composition approaches in the form of a hierarchical taxonomy. TCS provides a novel classification approach providing mainly four levels of hierarchy in classification taxonomy as shown in Fig. 1. The focus of classification at first level is the amount of user-involvement in the composition process. It provides three nodes: Manual/User-Defined, Semi-Automatic, and Automatic. The second level, which is based on the procedure or steps followed in the composition process, creates template and instance based categories under user-defined node, while Declarative, Workflow, Template/Instance, AI Planning [5], Ontology based, and Hybrid categories under Automatic node

[1]. The third level of taxonomy is based on the amount of dynamicity involved in the composition process and the technology used to implement the procedure adopted in second level. It creates dynamic and static nodes under each of template based, instance based, and workflow based nodes of second level; chaining, logic, and rule based under AI planning based; and context, and agent based under ontology based node of second level [4]. The fourth level of taxonomy produces the categories as shown in Fig. 1 and is focused on some variations possible in the technology adopted in the third level. We will use this classification system as the base for the implementation of TRS, which can be used to provide recommendations regarding the SWS composition approach to be used under some defined conditions.
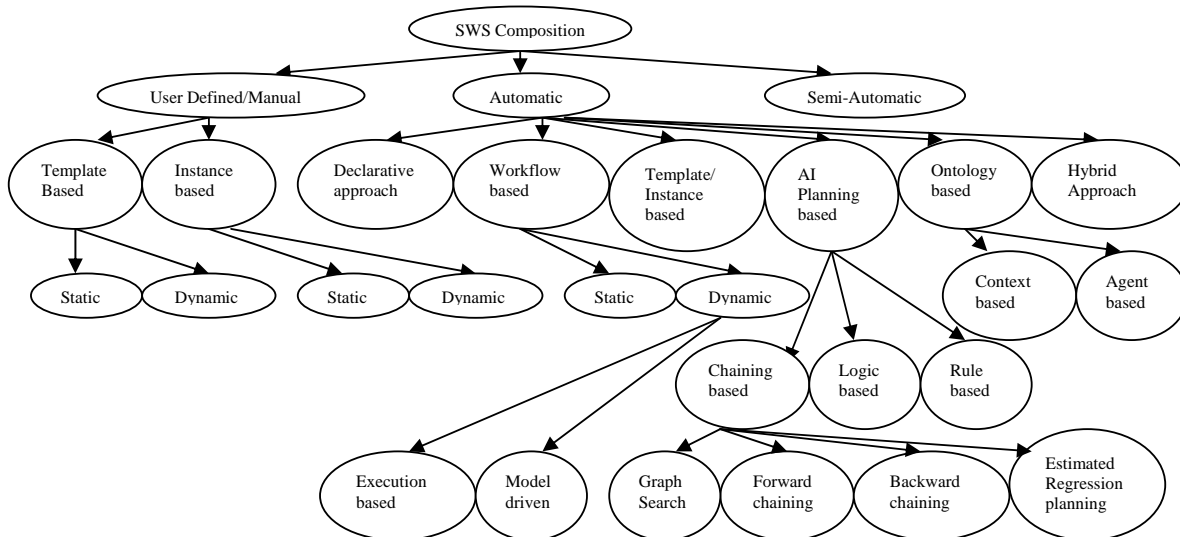

Fig. 1 A classification of SWS composition approaches

B. TRS Architecture

TRS uses rule-based, modular architecture for providing recommendations to the developers. The architecture of the TRS is shown in Fig. 2 in the form of block-diagram. As shown in figure, the basic architecture of the system can be divided into multiple modules, based upon the number of levels of abstraction it is providing. The fig. depicts it for two levels and it can be similarly extended. There are two ways of interacting with the system through Interface module: User Interface provides interaction of developer with the system, and Expert Interface provides interaction of the expert. The Questionnaire repository has various questions that are asked to the user regarding the semantic web based system to be developed. This repository is also accessible from the expert interface and expert has the authority to update or delete or modify any question from repository, however from the user interface, the questions are only observable and can be answered in a controlled way. TCS based rules repository has If … Then rules (as shown in Table I for example) which are framed based on the classification provided by TCS. The set of facts provides the facts about the recommendations for different possible values in the rules or different combination of multiple rules. In the explained architecture, these three parts are maintained separately for each level of abstraction. The Level-1 Module has parts for the first level of TCS or first and highest level of abstraction. The Level-2 Module has three parts for second level of TCS, which further has two sub-modules: Level-2U for User-Defined part of second level and Level-2A for Automatic part of second level. The answer given by the developer after placing in the appropriate rules from rule-repository are used by the rule-engine to compare them against the set-of-facts and design & provide the recommendations to the developer through recommendation display module. However, Questionnaire module, TCS based rule repository, and set of facts can also be maintained centrally for all the levels, but maintaining in a distributed manner provides faster processing of the system. It is due to the facts that when developer needs recommendations for a particular level, then for the centralized system, there will be need of searching the whole questionnaire repository for getting the appropriate questions and rule engine have to see from the relatively large rule-repository and set-of-facts. The user-learning module is used to teach the user about the necessary details to properly understand the recommendations provided by the system. For example, the understanding of TCS will enhance the user's ability to properly use the recommendations. The rule-learning module can be used by the system to learn new rules from the expert.
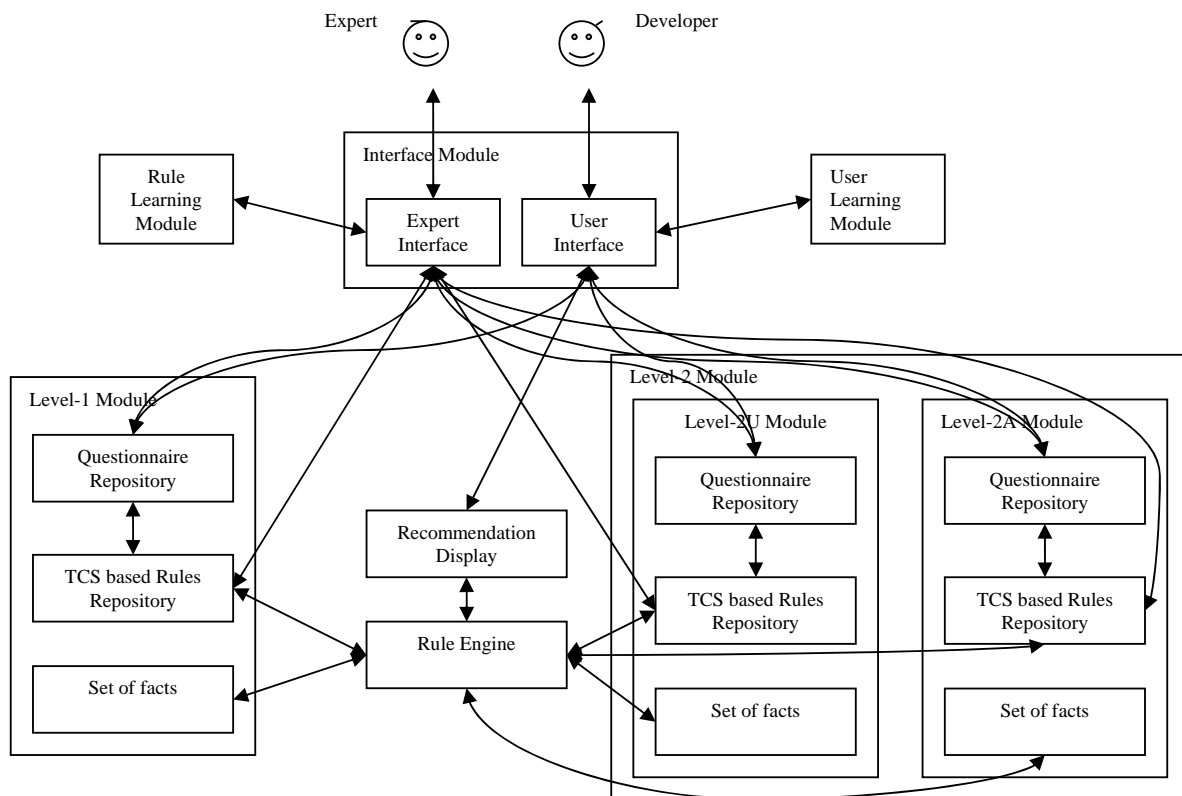
Fig. 2 TRS Architecture

III. EVALUATION AND IMPLEMENTATION

The production rules in the system are designed by keeping in mind the presented hierarchical classification (Table I).

For the first level of described taxonomy, we have designed about 28 rules. These rules cover all the three possible techniques at the first level of taxonomy. For the second level of taxonomy, we have implemented about 18 rules; 8 for user-defined node and 10 for automatic node. These rules provide recommendations regarding all the techniques at the second level of taxonomy. These rules, in addition of covering all techniques at first two levels, can also cover a large group of situations of use by generating combinations among each others. Due to space constraints, we have not provided detail of all the rules, but only some of sample rules are shown in Table I.

The system has modular architecture and is divided in modules based on the levels of hierarchy for which it is providing recommendations. The system thus is highly flexible and can be extended to any levels of taxonomy. It has an interface module for interacting with the user (Fig. 3).

TABLE I
SAMPLE RULE IN TRS

Rule 1: IF The user involvement in the composition process is very high THEN The use of manual/user-defined approach for service composition is very highly recommended.
Rule 2: IF Most of the input values to the services in targeted composite service are such that they highly vary to the taste of user and only decidable by the user of system THEN The use of manual/user-defined approach for service composition is very highly recommended.
Rule 3: IF The discovery and binding of services has to be performed dynamically at run-time THEN The use of automatic approach for service composition is very highly recommended.
Rule 4: IF Some of the new services may be needed while execution of the process THEN The use of automatic approach for service composition is very highly recommended.
Rule 5: IF Machine processble, highly accurate and real-time output is required THEN The use of automatic approach for service composition is very highly recommended.
Rule 6: IF The selection parameters for service components are decided at run-time THEN The use of automatic approach for service composition is very highly recommended.

.
.
.

The questions are stored in a questionnaire-module. Rule-engine designs recommendations by using the user's responses to the questions against specified rules, and the set-of-facts. A rule-learning module is also provided in the system which can only be accessed by an authorized expert. This module can be used to learn new rules in the system. The rules in the system are designed in such a way that they not only provide recommendations for a particular approach to use but also give indication on how much it is necessary to use the recommended approach. Rules 1 and 2 in Table I show the high necessity of following its recommendations.

Fig. 3(a) User Interaction Screen

Fig. 3(b) Recommendations for Fig. 3(a)

To the best of our knowledge, no work was found in the literature addressing the same idea as in TRS. However, the works by [1], [2], [3], [4], [5] provides classification of SWS composition approaches, and their works can be useful in this direction. But, all of these mentioned works provides a theoretical ground only and further only cover some specific categories of SWS composition approaches. On the other hand, the presented system, TRS, addresses the practical issues involved in choosing the composition approach to use. It covers a wide range of SWS composition approaches in its classification hierarchy, which is further used in the system architecture. Instead of providing only theoretical ground, the work also proposed a fully novel rule-based modular architecture for the TRS. A lot of implementation issues involved in the system have also been dealt. So, as compared to the existing work mentioned above, the presented architecture and the prototype can be easily extended by the perspectives users for their practical use.

IV. CONCLUSION

The paper presented modular, rule-based recommendation system, TRS, for SWS composition techniques. A taxonomical classification of SWS composition approaches has also been presented. Different issues involved in the implementation of TRS have been dealt. The work has also been evaluated and compared against the existing similar research and comparatively better features have been reported. The presented work can find the good use among the research and developer community. Our future work will involve firstly enhancing further the architecture of presented recommendation system and to develop and/or propose a semantic web service composition system based on the requirements and capabilities recognized in this work.

REFERENCES

[1] A. Alamri, M. Eid, A. E. Saddik, "Classification of the state-of-the-art dynamic web services composition techniques", *Int. J. web and grid services, 2*(2), 2006, pp 148-166.
[2] I. B. Arpinar, B. Aleman-Meza, R. Zhang, A. Maduko, "Ontology-Driven web services composition platform", *IEEE International Conference on e-Commerce Technology*, 2004, pp 46- 152.
[3] S. Dustdar, W. Schreiner, "A survey on web services composition", *Int. J. web and grid services*, *1*(1), 2005, pp 1-30.
[4] U. Kuster, M. Stern, B. Konig-Ries, "A Classification of Issues and Approaches in automatic Service Composition", *1st Int. Workshop on Engineering Service Compositions (WESC05) at ICSOC,05*, Amsterdam, Netherlands, 2005.
[5] J. Rao, X. Su, "A Survey of Automated Web service composition methods", *Proc: 1st international workshop of semantic web service and web process composition*, USA, 2004.

Sandeep Kumar is a research scholar with Department of Computer Engineering, Institute of Technology, Banaras Hindu University (IT-BHU), Varanasi, India-221005. He has done his BTech in information technology (Hons) and Gold Medal of the university and has completed his PhD course work in computer engineering with the best grade point from IT-BHU. He has many years of experience as a software engineer as well as teaching staff. He has presented several papers on national and international levels and has also authored multiple of books. His current areas of interest include Semantic Web, Web-based systems, multi-agent systems, knowledge-based systems, and software engineering.

R. B. Mishra (BSc, Eng.; MTech; PhD) is a reader with the Department of Computer Engineering, Institute of Technology, Banaras Hindu University (IT-BHU), Varanasi, India-221005. He has around 30 years of experience in teaching and research. He has published more than 80 research papers and articles in journals and conferences. He has supervised three PhD and 21 MTech dissertations. He has also visited as faculty to the University of Bath, UK. His current areas of interest include AI and its application to medicine, robotics, and the Semantic Web.