International Journal of Applied Information Management
Vol. 2, No. 1, April 2022, pp. 44-49 ISSN 2579-7069

Knowledge-Driven Automated Service Composition as a Method for Developing Decision Support Systems

Kiyohito Fujishima*

Ikuei Junior College 1656-1, Kyomemachi, Takasaki, Gunma, Japan
fujishima@ikuei-g.ac.jp *
* corresponding author

Abstract

A decision support system often necessitates a large amount of effort from both a domain modeling and technological standpoint. Using automated service composition, the study provides a method for minimizing the complexity of decision support system development. The justification for the approach is that many software systems (including decision support systems) today are based on service-oriented architecture, and the development of such systems may be described to some extent as a building composition of services that meet the required requirements. Automated planning algorithms can successfully address the difficulty of creating such compositions. The functional framework of decision support systems, requirements analysis for configurable service-oriented decision support systems and its primary components, and a conceptual model of a configurable service-oriented decision support system are all presented in the article.

Keywords: DSS, Knowledge Driven, Decision Support

1. Introduction

The creation of a decision support system (DSS) is a time-consuming and labor-intensive procedure. This effort is mostly made up of two types of activity. Building domain models is the first type, which is useful for explaining and forecasting decision-related occurrences. It necessitates subject knowledge as well as experience in a variety of applied mathematics methodologies. The second set of actions involves developing a technology solution that collects, processes, and displays data about the problem condition to the user (decision-maker). It primarily necessitates software engineering expertise. Reducing either type of work shortens the time it takes to design a DSS and lowers the cost.

The method described in this paper aims to reduce technical effort. Simplification of technical aspects of DSS development may allow a decision-maker to configure necessary components using high-level (and problem-oriented) tools, shortening the decision-making cycle (compared to the situation where each new decision of the decision-maker for information processing and presentation should be transferred to technical specialists and implemented). As a result, lowering the role of software engineers in the development of DSSs fulfills the demands of today's dynamic environment, which demands more and more speed from decision-makers in commercial and governmental organizations (see, e.g., [1]).

With the rise in popularity of service-oriented architecture (and, more recently, the microservice method [2]), the development of software systems (including DSSs) is increasingly seen as assembling existing services with desired functionality [3,4]. The building blocks for these compositions can come from either public service repositories or the organization's own repositories, but many of them aren't specific to the system in question. Furthermore, because of the low coupling (interface-based) characteristic of this type of software architecture, functional blocks can be implemented.

The research focuses on service-oriented DSSs and claims that employing automated service composition techniques, the work required to construct such DSSs can be reduced. The purpose is to provide a declarative tool for the decision-maker to express the restrictions of the needed composition. The physical implementation and invocation of the composition, as well as the identification of services that must be composed to meet these requirements, are all automated. The proposed approach is scientifically innovative in two ways. First, we use automatic service composition methods to construct DSSs while taking into account typical DSS designs and service structures. Then, we add conventional composition structures to existing automatic service composition algorithms, reducing the search space (during the composition).

To do so, we analyzed the common structures of service-oriented DSSs, the patterns they use, and the typical service compositions, and then used them in a formal framework. The rest of the paper is organized as follows. The second section discusses relevant research in the topic of automated service composition as well as the current state of the field. The functional framework of DSSs and the service composition patterns that are commonly utilized in them are described in Section 3. The basic requirements for service-oriented DSSs based on automated service composition are listed in Section 4. Section 5 concludes with a description of the proposed design model for such a system.

2. Literature Review

With the introduction of web services and service-oriented architecture, the idea of creating service compositions in an automated manner arose (see, e.g., [5]). Since then, the field has progressed significantly, and a number of formal models and methods for solving automatic service composition tasks have been proposed: automatic planning, in particular HTN planning [6], colored Petri nets [7,8], and various meta-heuristics (in particular, ant colonies, swarm intelligence, genetic algorithms) [9,10]. The Web Services Challenge - a forum where experience is exchanged on the development of automation tools for integrating web services - was held to debate and compare the outcomes produced by various research groups. [11].

Taking into account various scenarios of service execution when creating a composition (in particular, the possibility of errors) [12], taking into account the features of new service architectures edge computing, fog computing, and, as a result, taking into account the features of service placement when building a composition are all current problems in the field of automatic service composition of services.

Linguistic support is a crucial factor to consider when putting up service packages. For this goal, two primary groups of languages have been developed: the first group contains languages for "manual" compilation, such as BPEL and WS-CDL. These languages are mostly used by programmers to specify the composition in detail. OWL-S, a language for automatically composing services, is an example of the second group's languages. On the one hand, this language connects ontological modeling of service capabilities with modeling the creation of a trajectory in the state space typical of autonomous planning. This enables the use of automatic planning methods when putting together a composition of services specified using ontologies.

The strategy described in this research distinguishes itself by focusing on decision support systems and taking into consideration the characteristics of such systems when designing flexible service composition methods.

3. Research methods

In order to identify a typical functional framework of various types of DSSs, as well as to formulate requirements for configurable service-oriented DSSs, a literature analysis has been conducted. Traditionally, 5 types of DSS are distinguished [15]:

1) data-driven
2) model-driven
3) document driven
4) based on communication and group DSSs
5) knowledge-driven

We have decided to concentrate efforts, first of all, on model-driven and data-driven DSSs, because:

1) these types of DSSs are currently the most widely used
2) modern DSSs, for the most part, are complex, including elements of various “traditional” DSSs (the DICODE project [16] is quite indicative in this sense, simultaneously representing an explicit model of discourse and integrating data sources into this model – the first is typical for group DSS, and the second is for data-driven DSS)

Functional framework of a DSS of these types consists of the following blocks:

1) user interface and visualization components
2) data management
3) model management
4) solvers

The user interface and visualization components provide the interaction of different categories of users (end-user decision-maker, administrator and others) with the DSS. The role of visualization components [17] is quite important, as visual analysis of data and the results of applying models is one of the important sources of information for decision makers.

Data management is a multi-level process, including data collection (connecting to external data sources), their merging and coordination (integration and harmonization of data received from different sources, bringing them to the same format both at syntactical and semantic levels [18]), as well as analytical processing (obtaining derivative characteristics required by the decision maker). The specific operations carried out at the above stages are the application of functional transformations and operations of relational algebra to data sets.

Model management is based on the life cycle of a model in DSS and includes the following operations: creating a model, changing a model, linking the model with data, applying a solver to the model, integrating models (creating new models based on existing ones), deleting the model. Solvers are functional blocks that implement certain algorithms for searching the values of independent model variables corresponding to optimal or effective values of decision quality functions given by the decision maker.

A typical pattern used in data-driven DSS is the construction of an acyclic transformation graph, in the nodes of which are operations for loading and transforming data, and arcs determine the sequence of operations [19]. When constructing model-based DSS, typical patterns are [3]: encapsulating a problem-oriented model in the form of a service, associating models with data, models with a solver, various methods of complexing models (aggregation (combining several models into one), classification (several models are varieties of one), sequential combination (the output of one model gets to the input of another).

4. Analysis and discussion of results

Two types of information sources were considered while establishing the list of requirements for configurable service-oriented decision support systems (SO DSSs):

1) The findings of a literature review conducted in the field of current DSSs (including service oriented DSSs)
2) The analysis of existing automatic composition techniques and the basic data required for them.

There are two types of requirements that have been developed: a) general (basic) requirements for DSSs (related to the contents of SO DSSs and assuring compatibility with the existing technological stack of service-oriented systems); b) particular requirements for SO DSSs based on automatic service composition (due to the need to support automatic composition algorithms). DSSs must meet the following broad (basic) standards. A DSS should enable decision makers to quickly "penetrate into the essence" of available data by executing a collection, integration, transformation, discovery, and training cycle [3]. To do so, you'll need to:

1) DSSs should be adaptable in order to give access to diverse resources that may be necessary during decision support [20] and to deal with changing (and fluctuating) decision requirements from one interested person to another.

ISSN 2579-7069

2) Iterative methodologies (for example, what-if analysis) should be available in DSSs to assist users in refining and customizing models.
3) Changes in the external, internal, and system environments should be able to modify DSSs.
4) Flexible manipulations with components and processes should be supported by DSSs (for example, there should be no restrictions on the part of the system on the choice of some components to support decision-making or access to some resources). In addition, the architecture should allow for the real-time addition of additional components to the system.

The usage of a service-oriented design considerably simplifies the implementation of these criteria, as can be seen. Specific requirements for SO DSSs that are composed automatically:

1) The prospective components of a DSS (services) must be accompanied with (machine readable) information to enable automatic composition. These metadata are also required for acquiring access to sources, developing data integration schemes and merging models, customizing the user experience, and solving other issues. The usage of ontologies is a well-known method of displaying such data. As a result, principles for (a) ontological description of various groups of services (in accordance with the functional framework), (b) ontological description of tasks, and (c) determining the possibility of service composition using the ontological description of services and tasks should be developed.
2) The user interface and visualization components, which lie between the DSS user and the data processing and model execution components, must be customizable so that the results of any services can be displayed.

5. The proposed design of a configurable service-oriented decision support system

A conceptual model of a configurable service-oriented DSS is proposed in accordance with the identified functional framework (Figure 1).

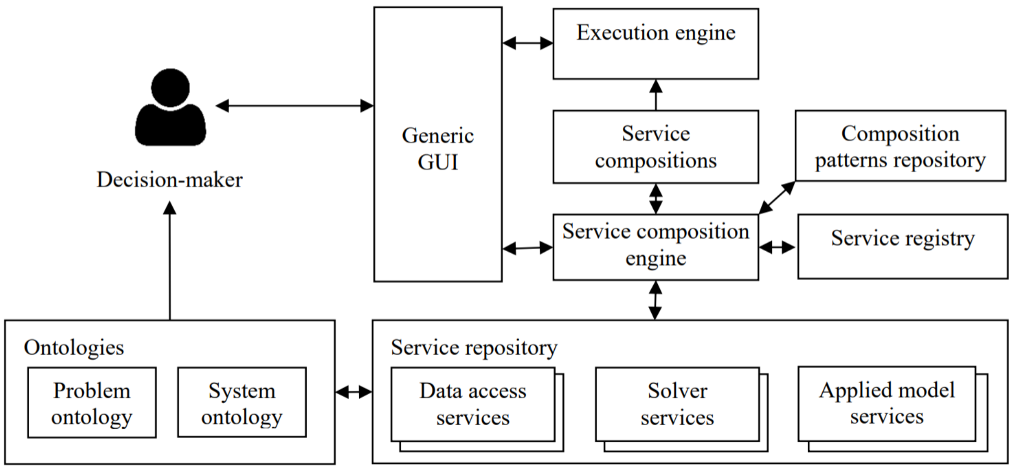

Figure. 1. Conceptual model of a DSS based on automatic service composition

Services are the foundation of the SO DSS. Data access services, solver services, and applied model services are all examples of services. Each service has an ontological description, which includes a description of the input and output, as well as quantitative features of the service, in order to execute the capabilities of semantic search and subsequent automatic compilation (required time, resources). Two ontologies are used to create the ontological description: a problem-oriented ontology (created for a specific DSS application area) and a system ontology (used primarily by automatic composition algorithms and based on OWL-S ontology [21,22]). Composite services, which are generated from other services at the request of a decision maker utilizing specified composition processes (for example, coupling the output parameters of one service to the input parameters of another) in automatic or manual mode, are a special sort of service. All SO DSS services are registered in a special registry and can be used to create composite services.

The core of the SO DSS is the infrastructure for creating composite services, which can be combined to form a new composite service (or several options offered by the decision maker) based on the decision maker's ontology (domain and system ontology) and the descriptions of services registered in the registry. Service kinds are considered in the composition process to decrease the search space (based on the functional structure of DSSs). A typical composite service will therefore contain various data access services, an applied model service, and a solver service, to which model-data binding and model-solver will be applied (see the patterns listed in section 3).

The research's uniqueness and novelty, particularly the suggested conceptual model of the SO DSS, arises from the fact that a large portion of current DSS research is focused on establishing specialized models of decision support in certain topic areas [23,24], with architecture difficulties being secondary. The functional composition of the system is considered constant and is stated in the design process in those publications where architecture development challenges are still in the spotlight. Many recent approaches for synthesizing such systems based on requirements are similar in that the end product is still a system with "fixed" functionality that isn't always able to adapt to changing situations. Nonetheless, the list of the most significant needs for DSSs frequently includes extensibility, adaptability to new situations, and simplicity of execution of complex analytical scenarios without the involvement of software development specialists [20].

These needs can be taken into account using the conceptual model that has been constructed. It allows one to expand them by integrating different classes of DSSs and providing the possibility of automatic composition of services in accordance with the tasks of the decision-maker (see, for example, [3,4]). It is compatible (in functional and architectural ways) with modern developments in the field of DSSs (see, for example, [3,4]).

6. Conclusion

The paper addresses the problem of reducing the complexity of decision support systems development. To achieve that, it proposes an approach to create decision support systems based on automated service composition. According to the proposed approach, a decision-maker should specify constraints of the required composition with a declarative tool, while the identification of services that have to be composed in order to satisfy these constraints, as well as physical implementation and invoking of the composition are implemented automatically. This approach reduces the cost of the development of a DSS, speeds up the development, and allows to shorten the decision-making cycle. Specifically, the paper identifies functional framework of different kinds of DSSs (to use it for service annotation in order to reduce the complexity of finding the composition), summarizes the requirements for service-oriented DSSs and describes a design of the service-oriented DSS, based on automatic service composition.

References

[1] Alemany M M E, Boza A, Ortiz A and Fuertes-Miquel V S 2016 Configurable DSS for Uncertainty Management by Fuzzy Sets Procedia Computer Science 83 1019-24

[2] Dragoni N, Giallorenzo S, Lafuente A L, Mazzara M, Montesi F, Mustafin R and Safina L 2017 Microservices: Yesterday, Today, and Tomorrow Present and Ulterior Software Engineering (Cham: Springer International Publishing) pp 195-216

[3] Dong C-S S J and Srinivasan A 2013 Agent-enabled service-oriented decision support systems Decision Support Systems 55 364-73

[4] Nada A, Nasr M and Salah M 2014 Service oriented approach for decision support systems 2014 IEEE 7th Joint International Information Technology and Artificial Intelligence Conference pp 409-13

[5] Peer J 2004 A PDDL Based Tool for Automatic Web Service Composition Principles and Practice of Semantic Web Reasoning. Lecture Notes in Computer Science 3208 149-63

ISSN 2579-7069

[6] Georgievski I, Nizamic F, Lazovik A and Aiello M 2017 Cloud Ready Applications Composed via HTN Planning IEEE 10th Conference on Service-Oriented Computing and Applications (SOCA) (IEEE) pp 81-9

[7] Kalamegam P 2017 Usage of CPN Models in Web Service Compositions - A Survey International Conference on Technical Advancements in Computers and Communications (ICTACC) (IEEE) pp 4-6

[8] Li K, Li W, Sun X and Xia Y 2016 A Stochastic-Petri-Net-Based Model for Ontology-Based Service Composition 9th International Conference on Service Science (ICSS) (IEEE) pp 108– 12

[9] Gil-Herrera J and Botero J F 2017 A scalable metaheuristic for service function chain composition IEEE 9th Latin-American Conference on Communications (LATINCOM) (IEEE) pp 1-6

[10] Dahan F, El Hindi K and Ghoneim A 2017 An Adapted Ant-Inspired Algorithm for Enhancing Web Service Composition International Journal on Semantic Web and Information Systems 13 181-97

[11] Bansal A, Bansal S, Blake M B, Bleul S and Weise T 2012 Overview of the Web Services Challenge (WSC): Discovery and Composition of SemanticWeb Services Semantic Web Services pp 297-311

[12] Zhu W, Bastani F, Yen I-L, Fu J and Zhang Y 2017 Automated Holistic Service Composition: Modeling and Composition Reasoning Techniques IEEE International Conference on Web Services (ICWS) (IEEE) pp 596-603

[13] Al Rishawi I, Kotb Y and Al Ridhawi Y 2017 Workflow-Net Based Service Composition Using Mobile Edge Nodes IEEE Access 5 23719-35

[14] Qian W, Peng X, Sun J, Yu Y, Nuseibeh B and Zhao W 2017 O2O service composition with social collaboration 32nd IEEE/ACM International Conference on Automated Software Engineering (ASE) (IEEE) pp 451-61

[15] Power D J 2002 Decision Support Systems: Concepts and Resources for Managers (Praeger)

[16] Karacapilidis N 2014 Mastering Data-Intensive Collaboration and Decision Making (Cham: Springer International Publishing) 5

[17] Lengler R and Eppler M J 2007 Towards a periodic table of visualization methods of management Graphics and Visualization in Engineering ed M Alam (Anaheim: ACTA Press) pp 83-8

[18] Eine B, Jurisch M and Quint W 2017 Ontology-Based Big Data Management Systems 5 45

[19] Anon Apache Airflow

[20] Argyris C 1976 Single-loop and double-loop models in research on decision making Administrative Science Quarterly 21 363-75

[21] Anon OWL-S: Semantic Markup for Web Services

[22] Martin D, Paolucci M, McIlraith S, Burstein M, McDermott D, McGuinness D, Parsia B, Payne T, Sabou M, Solanki M, Srinivasan N and Sycara K 2005 Bringing Semantics to Web Services: The OWL-S Approach pp 26-42

[23] Kandakoglu A, Sauré A, Michalowski W, Aquino M, Graham J and McCormick B 2019 A decision support system for home dialysis visit scheduling and nurse routing Decision Support Systems 113224

[24] Gardas B B, Raut R D, Cheikhrouhou N and Narkhede B E 2019 A hybrid decision support system for analyzing challenges of the agricultural supply chain Sustainable Production and Consumption 18 19-32