An approach to the development of decision support systems with knowledge-driven automated service composition

N Mustafin¹, A Ponomarev¹ and P Kopylov¹
¹ St. Petersburg Federal Research Center of the Russian Academy of Sciences (SPC RAS), 39 14th Line, 199178, St. Petersburg, Russia
E-mail: ponomarev.a.v@gmail.com

Abstract. The development of a decision support system typically requires a significant effort from both domain modelling and technical perspectives. The paper proposes an approach for reducing the complexity of decision support system development leveraging automated service composition. The rationale behind the approach is that today many software systems (including decision support systems) are based on service-oriented architecture and to some extent the development of such systems can be represented as building composition of services satisfying the specified requirements. The problem of building such compositions can be successfully addressed by automated planning algorithms. Particularly, the paper presents the functional framework of decision support systems, requirements analysis for configurable service-oriented decision support systems and their main components, and a conceptual model of a configurable service-oriented decision support system.

1. Introduction
The development of a decision support system (DSS) is a complex process requiring significant effort. This effort basically stems from two kinds of activities. The first kind is building domain models, helpful in explaining and predicting decision-related phenomena. It requires domain knowledge as well as expertise in a wide range of methods of applied mathematics. The second kind of activities is related to building a technological solution collecting the data about the problem situation, processing it, and presenting the results to the user (decision-maker). It requires mostly expertise in software engineering. Reducing either kind of effort speeds up the development of a DSS and reduces the cost of the development.

The approach proposed in this paper focuses on reducing the technical effort. Simplification of technical aspects of DSS development may give a decision-maker the ability to configure necessary components using high-level (and problem-oriented) tools, allowing to shorten the decision-making cycle (compared to the situation when each new decision of the decision-maker for the processing and presentation of information should be transferred to technical specialists and implemented by them) in addition to already mentioned effects of speeding up the development and reducing the cost of a DSS. Therefore, reducing the role of software engineers in development of the DSSs meets the requirements set by modern dynamic environment demanding more and more speed from decision-makers managing commercial organizations and governmental agencies (see, e.g., [1]).

With the growing popularity of the service-oriented architecture (and, more recently, the microservice approach [2]), the development of software systems (including DSSs) often can be viewed as building compositions of existing services with required functionality [3,4]. The building blocks for these compositions can either be taken from public service repositories, or from the
organization’s own repositories, but many of them are not specific to the particular system being constructed. Besides, low coupling (interface-based), characteristic for this kind of software architectures allows to easily change implementation of functional blocks.

The paper focuses on service-oriented DSSs and proposes to reduce the effort of building such DSSs by using automated service composition techniques. The goal is to give the decision-maker a declarative tool that could be used to specify the constraints of the required composition. 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. The scientific novelty of the proposed approach is twofold. First, we apply the methods of automatic service composition to the development of DSSs, taking into account typical architectures and service structure of DSSs. Then, we extend existing automatic service composition methods with typical composition structures, allowing to reduce the search space (during the composition).

To achieve this, we have performed the analysis of typical structures of service-oriented DSSs, patterns used in them, and typical service compositions, and then, utilize them in a formal framework.

The rest of the paper has the following structure. Section 2 describes related research in the field of automated service composition and the status of this research area. Section 3 describes functional framework of DSSs and service composition patterns typically used in them. Section 4 contains main requirements for service-oriented DSSs based on automated service composition. Finally, Section 5 describes the proposed design model of such system.

2. Related work
The idea of building service compositions in an automated way first appeared with the advent of web services and service-oriented architecture (see, e.g., [5]). Since that time, the area has advanced significantly and a number of formal models and methods has been proposed to solve the tasks of automatic service composition: automatic planning, in particular, HTN planning [6], colored Petri nets [7,8], various meta-heuristics (in particular, ant colonies, swarm intelligence, genetic algorithms) [9,10]. To discuss and compare the results achieved by various research groups, The Web Services Challenge was organized - a forum where experience is exchanged on the development of automation tools for integrating web services [11].

Current problems in the field of automatic service composition of services are 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 the placement of services when building a composite [13,14].

An important aspect in building service compositions is linguistic support. Currently, two main groups of languages have been developed for this purpose: the first group includes languages for "manual" composition - BPEL, WS-CDL, etc. These languages are used mainly by programmers for a detailed specification of the composition. A representative example of the languages of the second group is OWL-S, designed for automatic composition of services. This language provides a link between ontological modelling of the service capabilities, on the one hand, and modelling the construction of a trajectory in the state space typical for automatic planning, on the other hand. This allows applying automatic planning methods when constructing a composition of services described with a help of ontologies.

The distinguishing feature of the approach proposed in this paper is the orientation on decision support systems and taking into account the specifics of such systems when developing ways of flexible composition of services.

3. Functional framework of decision support systems and typical patterns used in them
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: a) these types of DSSs
are currently the most widely used, b) 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: a) user interface and visualization components, b) data management, c) model management, d) 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. Requirements analysis
In forming the set of requirements for configurable service-oriented decision support systems (SO DSSs), two classes of information sources were taken into account:

1) The results of the conducted literature review in the field modern DSSs (including service-oriented DSSs);
2) The analysis of existing algorithms for automatic composition and the initial data necessary for them.

The set of developed requirements can be divided into two groups: a) general (basic) requirements for DSSs (associated with the contents of SO DSSs and ensuring their compatibility with the existing technological stack of service-oriented systems), b) specific requirements for SO DSSs based on automatic composition of services (due to the need to support automatic composition algorithms).

There are following general (basic) requirements for DSSs. A DSS should allow decision makers to quickly “penetrate into the essence” of the available data, implementing a cycle of collection, integration, transformation, discovery and training [3]. In order to achieve this:

1) DSSs should be adaptive in order to provide access to various resources that may be required during decision support [20] and deal with changing (and varying from one interested person to another) decision requirements;
2) DSSs should provide iterative methods (for example, what-if analysis) to help users refine and customize models;
3) DSSs should be able to adapt to changes in the external, internal and system environments;
4) DSSs should support flexible manipulations with components and processes (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). The architecture should also allow the addition of new components to the system in real time.

It can be noticed, that the use of a service-oriented architecture greatly facilitates the implementation of these requirements.

Specific requirements for SO DSSs based on automatic composition:

1) To enable automatic composition, the potential components of a DSS (services) must be accompanied by (machine readable) metadata. These metadata are also necessary for gaining access to sources, building data integration schemes and merging models, adapting the user interface and solving other problems. A well-established approach to presenting such metadata is the use of ontologies. Accordingly, principles should be developed for (a) ontological description of different groups of services (in accordance with the functional framework), (b) ontological description of tasks, and also (c) determining the possibility of service composition taking into account the ontological description of services and tasks.

2) The user interface and visualization components, occupying a position between the DSS user and the data processing and model execution components themselves, must be configurable to provide the ability to display the results of any services.

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

In accordance with the identified functional framework, a conceptual model of a configurable service-oriented DSS is proposed (Fig. 1).

![Conceptual model of a DSS based on automatic service composition.](image)

The basic elements of the SO DSS are services. There are several types of services: data access services, solver services, applied model services. To implement the capabilities of semantic search and subsequent automatic composition, each service has an ontological description, including a description of the input and output, as well as quantitative characteristics of the service (required time, resources). The ontological description is formed in terms of two ontologies: a problem-oriented ontology (developed for a particular application area of the DSS) and a system ontology (used primarily by automatic composition algorithms and based on OWL-S ontology [21,22]). A special type of service is composite services formed from other services using certain composition operations (for example, linking the output parameters of one service and the input parameters of another) at the request of a decision maker in automatic or manual mode. All services of the SO DSS are registered in a special registry and are available for building composite services.
The core of the SO DSS is the infrastructure for building composite services, which can form a new composite service (or several options that are offered by the decision maker), based on the specification of the problem formulated by the decision maker in terms of ontology (domain and system ontology) and the descriptions of services registered in the registry. To reduce the search space in the process of composition, service types are taken into account (based on the functional structure of DSSs). Therefore, a typical composite service will include several 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 originality and novelty of the research and, in particular, the proposed conceptual model of the SO DSS, lies in the fact that a significant part of modern research in the field of DSSs is aimed at developing specific models of decision support in certain subject areas [23,24], the architecture issues of such systems are secondary. In those publications where architecture development issues are still in the focus of attention, the functional composition of the system is considered constant and is specified in the design process. The same applies to many modern methodologies for synthesizing such systems based on requirements – the result of this synthesis is still a system with “fixed” functionality, not always able to adapt to changing conditions. Nevertheless, extensibility, adaptability to new conditions, and ease of implementation of complex analytical scenarios without the participation of software development specialists are often included in the list of the most important requirements for DSSs [20].

The developed conceptual model allows taking these requirements into account. Being compatible (in a functional and architectural ways) with modern developments in the field of DSSs (see, for example, [3,4]), it allows one to expand them by integrating different classes of DSSs and by providing the possibility of automatic composition of services in accordance with the tasks of the decision-maker.

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.

7. Acknowledgments

The reported study was funded by RFBR, project number 19-07-00928.

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

[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 Ridhawi 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, Burstine 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] Kandakoglou 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, Cheikhouhou 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