## SERVICE COMPOSITIONS IN PROBLEMS OF URBAN PLANNING

Roman K. Fedorov<sup>(1)</sup>, Alexander S. Shumilov<sup>(2)</sup>

<sup>(2)</sup> Irkutsk Scientific Center SB RAS, Irkutsk, Russia
<sup>(2)</sup> Matrosov Institute for System Dynamics and Control Theory SB RAS, Irkutsk

Information technologies are being applied in almost all fields of human activities, including the urban planning. One of the common problems in urban planning is the estimation of transport and walking availability of various socially important facilities. Geoportal of ISDCT SB RAS proposes to solve this problem using service compositions, defined in a form of JavaScript programs. During the composition execution, the automatic parallelization of services is performed in order to minimize the overall composition execution time. The scenario planning and execution processes are shown, as well as the ability to adapt to changes in computing environment.

Keywords: service compositions, task scheduling, distributed computing.

**Introduction.** The volume of received and generated data constantly increases in the world of information technologies. In order to process data the service-oriented approach (SOA) is used more and more often [1]. SOA proposes to publish computational algorithms and data sources in a form of services, available on the Internet. The developments of SOA principles are service compositions – defined interactions between existing services that solve specific complex problem. As services can be launched on multiple computational nodes at the same time, the problem of parallelization of services arises in order to minify the overall service composition execution time. This work considers the problem of estimating the transport and walking availability for health facilities within the task of urban planning.

One of the generally recognized standards of service interaction definition in compositions is the usage of the directed acyclic graph (DAG) [2]. Vertices of the graphs are service calls, edges are data dependencies between services. All algorithms that deal with the DAG scheduling can be divided in two categories according to the way the schedule is accomplished – heuristic and meta-heuristic algorithms.

One of the most common DAG scheduling algorithms is the Heterogeneous Earliest Finish Time (HEFT) [3]. The most popular class of algorithms among meta-heuristic ones is the class of the genetic algorithms [4]. They usually find the approximate solution from the large search space by applying the evolutionary algorithms. Genetic algorithms usually find more precise solutions than heuristic algorithms; however, they take a lot more time to complete [5].

Works that propose hybrid scheduling algorithms started to appear lately. These algorithms combine two or more DAG scheduling approaches. For instance, the Hybrid Evolutionary Work-flow Scheduling Algorithm [6] uses both HEFT algorithm and genetic approach. This algorithm is also able to adapt to changes that can occur in the environments, which is very important when it comes to scheduling in real distributed heterogeneous systems.

Talking about ways to define the service composition, it is essential to consider the two most common approaches – using the BPEL (Business Process Execution Language) and XPDL (XML Process Definition Language) standards. These approaches define the DAG in descriptive manner. Despite the fact that the XPDL standard was developed earlier, the BPEL is used more often be-

cause it is more suitable for describing service interactions. However, XPDL allows calling of services that do not have standardized web-interface, also it allows describing the graphical view of the defined interactions. Another promising way of defining service compositions is using the programming language, where service calls are performed by calling corresponding functions. Defining compositions in form of programs allows processing intermediate result in the actual composition code, as well as makes it possible to apply existing software packages for specific programming language.

One of the most frequently used software packages for defining the service compositions using the BPEL standard is the Oracle BPEL Process Manager. This commercial software product has graphical user interface for composition creation and debugging, it support large number of various web-service interface standards (WSDL, WPS, etc.). BPEL Process Manager is not capable of service scheduling, which is critical when it comes to planning of complex service compositions with a large number of participating services.

Another popular software product that deals with the service composition is the Apache Tavern, which lets defining the composition using the set of graphical primitives and special user interface. Unlike the imperative BPEL that requires the explicit definition of execution process, Tavern uses the functional model managed by data. Tavern is well known among scientific teams that carry out research in fields of astronomy and bio-informatics (for example, for detecting genes that are responsible for specific disease). However, Taverna does not perform task scheduling as well.

Thereby, there are existing approaches to service compositions definition, where DAG is completely defined before the actual composition execution starts, as well as algorithms that minimize the overall composition execution already exist. However, there are no algorithms, approaches or software packages that allow automatic scheduling of the service composition defined in a programming language in the distributed heterogeneous computational environment.

As an approbation of the proposed and developed system of service composition the task of transport and walking availability estimation of various socially important facilities was solved. Such estimations play important role in various urban development procedures – planning public transportation, location of socially important objects, road networks and theirs throughput etc. While performing estimations it is important to consider the availability both by transport and by foot, as well as to take into account various natural barriers and overpasses (for example, rivers and bridges). In order to estimate the availability of various infrastructure objects the set of services and theirs composition was developed. All of services are deployed on a number of computational nodes of the local cloud infrastructure. The actual service composition that solves the problems is described below.

**Distributed services composition environment.** The distributed services composition environment is the software system that allows organizing the service composition execution defined using the JavaScript programming language. The environment is integrated with Geoportal – web-application that was developed in ISDCT SB RAS and that provides the set of instruments for spatial data processing. Main components of the environment are the composition development, parameters input and composition execution web-interfaces and the service dispatcher.

Service composition development web-interface allows creating composition in a form of scenarios in JavaScript programming language using the special web-form, which simplifies and corrects the scenario development process. Before developing the actual scenario code user needs to define input and output parameters of the scenario. Within the solution of task, stated above, first parameter will be vector files with point infrastructure objects in the SHP spatial data format. The second parameter is the SHP file with linear objects that define natural boundaries (water bodies). The third parameter is the SHP file that contains linear object that correspond to road network. The last parameter is the extent of computation (the estimations will be computed only for those object that are inside of the defined rectangle). The output parameter defines the folder that will store result of the composition execution that will be downloaded from the remote server.

After definition of input and output parameters user defines the service composition code. Special text field with JavaScript syntax highlighting is available while editing the scenario code, as well as the list of available services.

Four services are used in the composition:

*shp\_to\_geojson\_converter* – converts SHP file that stores the point objects data into the GeoJSON one. Resulting GeoJSON is stored as a regular string and is lately parsed and processed in the JavaScript scenario;

*geojson\_to\_shp\_converter* – converts GeoJSON data into the SHP file. GeoJSON object that was received after completion of the previous service is parsed and split in separate objects, then every split point object is converted back into the SHP file, so it can be processed by next service;

*bufferize\_vector* – creates buffer zones around point objects of the provided SHP file. Results of the previous service are converted from point to polygon ones with specific radius of buffer;

*road\_analysis* – estimates the transport and walking availability for point objects according to the information about road network and natural bounds. Estimations are done for objects that are results of the previous service execution. The result of the service is the regular grid in raster spatial data format GeoTIFF, which contains average time to reach in every grid cell for specific health facility.

*The service composition dispatcher* accepts the scenario code as an input, as well as parameters entered by user, and starts the execution. During scenario execution dispatcher does following:

1. Builds directed acyclic graph of service call dependencies that can change during the composition execution;

2. Performs the scheduling of services that are called inside of the scenario according to changes that occur in the distributed computational environment;

3. Keeps track of the computational environment state and re-schedules services call plan when certain events occur in the environment.

During service composition execution the problem of services call planning arises. The problem is to assign service calls to such computational nodes at certain time, so that the overall duration of composition execution will be as low as possible. In order to find such schedule, that will ensure the minimal service execution time, the modification of list-based heuristic scheduling algorithm HEFT (Heterogeneous Earliest Finish Time) is used. The algorithm sorts tasks according to calculated priorities and cuts off the unpromising branches of solutions in order to speed up the schedule searching process. After the start of scenario execution the algorithm uses the information about current state of the computational environment in order to re-build the already executing part of the schedule [7].

The computational environment of the distributed services is heterogeneous, i.e. computational nodes, where services are deployed, have different software and hardware characteristics. Computational nodes can change their online/offline state, as well as changes in the network infrastructure, that provides access to computational nodes, can occur. Thereby, the scheduling algorithm has to rebuild the current schedule and take into account current execution state of the schedule when following events occur: computational nodes are added or deleted, the task in added to DAG, task that already completed took longer or shorter time to execute than planned, executing task ended with an error.

**Approbation**. In order to estimate the transport and walking availability for healthcare facilities the corresponding scenario has to be executed with following parameters:

• *hospitals* parameter accepts the Geoportal table, which stores the locations of healthcare facilities. The table data is converted into the SHP file. Estimations are calculated for objects, that are limited by extent, which can be defined using the rectangle tool in the interactive webpage map.

• *barrier* and roads parameters are SHP files that contain road network and natural barriers linear object.

• *extent* parameter defines the calculation area of estimations.

The result of the scenario execution is the set of raster images. Each of images contains the estimations for specific healthcare facility. Generated raster files can be displayed on the interactive web-map of the Geoportal.

The automatically generated DAG is displayed on the fig. 1. For example, the task with the 1001 identifier corresponds to the *shp\_to\_geojson\_converter* service call, so its results are required in order to call *geojson\_to\_shp\_converter* services in the loop. *geojson\_to\_shp\_converter*, *bufferize\_vector* and *road\_analysis* are called in the loop, which corresponds to the task sequences 1002 - 1003 - 1004, 1005 - 1006 - 1007 etc.



Figure. 1. Directed acyclic graph of task data dependecies

During the scenario execution the automatic DAG completion occurs. When the scenario completes, its results are uploaded to the Geoportal and displayed on the webpage using the interactive web-map. The fig. 2 contains the screenshot of the estimations visualizations for two different healthcare facilities. Using the visualization instruments of the Geoportal, the displayed raster images were colored according to availability intervals 0-10 minutes, 10-20 minutes, etc. It is worth noting, that resulting images contain empty cells which correspond to the natural barriers (water bodies particularly).



Figure. 2. Visualization of transport and walking availability estimations for healthcare facilities

**Conclusion.** The problem of transport and walking availability estimation for healthcare facilities for urban planning was solved as a result of applying the developed approach to service composition definition and execution in the distributed heterogeneous environment. The specific set of web-services was developed, as well as their composition in a form of scenario in the JavaScript programming language in order to solve this problem. The developed service scenario was executed within the distributed heterogeneous environment. During composition execution the adaptability of execution process to changes that occur in the computational environment was tested. The task solution was performed using the tools provided by Geoportal of ISDCT SB RAS.

Main advantages of the developed system for service composition execution is the adaptability to changes that occur in the computational environment, automatic scheduling of service calls within compositions and usage of the JavaScript programming language in order to build interactions between services for solving complex interdisciplinary tasks.

The work was carried out with financial support: Russian Foundation for Basic Research (grants  $N_{2}$ : 18-07-00758-a, 17-57-44006-MoHr\_a, 16-07-00411-a, 16-07-00554-a, 17-47-380007-p); RAS Presidium Program  $N_{2}27$ ; Integration program SB RAS, ISC SB RAS. Results are achieved using the Centre of collective usage «Integrated information network of Irkutsk scientific educational complex» (http://net.icc.ru).

## REFERENCES

[1] Mestre M.R., Ballesteros A.M. Analysis and design for the development of software with service-oriented approach: Characteristics and methodological framework // 2015 10th Computing Colombian Conference (10CCC), Bogota. 2015. P. 63-70.

[2] Selvi S., Manimegalai D. DAG Scheduling in Heterogeneous Computing and Grid Environments Using Variable Neighborhood Search Algorithm, Applied Artificial Intelligence. Vol. 31. 2017. P. 134-173.

[3] Zhao H., Sakellariou R. An Experimental Investigation into the Rank Function of the Heterogeneous Earliest Finish Time Scheduling Algorithm // Lecture Notes in Com-puter Science. – 2003. Vol. 2790. C. 189-194.

[4] Sinnen O. Task scheduling for parallel systems // Wiley-Interscience, 2007. P. 108.

[5] Omara F., Arafa M. Genetic algorithms for task scheduling problem // Journal of Parallel and Distributed Computing, 2010. Vol. 70, Is. 1. P. 13-22.

[6] Nasonov D., Butakov N., Balakhontseva M., Knyazkov K., Boukhanovsky A.V. Hybrid Evolutionary Workflow Scheduling Algorithm for Dynamic Heterogeneous Distributed Computational Environment // Advances in Intelligent Systems and Computing. 2014. Vol. 299. C. 83-92

[7] Федоров Р. К., Бычков И. В., Шумилов А. С., Ружников Г. М. Система планирования и выполнения композиций веб-сервисов в гетерогенной динамической среде // Вычислительные технологии. 2016. Т. 21, № 6. С. 18 – 35.