Workflows Generation based on the Domain Ontology

Larysa Globa, Maryna Popova, Andrii Bakai
Information and Telecommunication Department
Institute of Telecommunication Systems
National Technical University of Ukraine «Igor Sikorsky Kyiv Polytechnic Institute»
Kyiv, Ukraine
lgloba@its.kpi.ua, pma1701@gmail.com, andruha.bakay1997@gmail.com

***Abstract*—The digital economy, which is a prerequisite for the 4th Industrial Revolution, requires a rapid increase in services provided in the digital global space. However, the level of computer-aided workflows underlying the various services is imperfect due to the lack of meta-models (ontologies) that would describe the conditions of use of the services, take into account the meaningful connections between them, and a mathematical apparatus that would make it easy to execute certain interactions of elements of such models.**

The paper describes the approach to automation of the design of workflows and their components (functional services, communications and rules of interaction) by computer-aided generation of both a set of services that are components of workflows and the sequence of their execution, using ontologies – meta-models of workflows, services, domain, logical rules that establish relationships between functional services.

The application of the proposed approach allowed to generate workflows based on the parameters defined in the meta-model (ontology) of the register of services and limitations imposed by the domain ontology, taking into account both functional and non-functional requirements, realizing the transformation of the workflow in its program execution models regardless of the calculation.

For the computer-aided workflows generation, a software environment has been developed, the performance of which is tested on the example of workflow generation with the use of IT TODOS tools for constructing and modifying ontologies and generating a workflow scheme using BPMN.

***Keywords*—ontology, workflow, functional services, computer-aided business process design.**

I. Introduction

Nowadays is characterized by the rapid development of information technology, which transforms all spheres of society. The emergence of the digital economy in the industries and business has been called the 4th Industrial Revolution – Industry 4.0 [1], which has created a virtual (digital) business environment, which is constantly increasing the volume of services.

Under such conditions, technological innovation and reconfiguration of complex software solutions to adapt to the requirements of the global business environment are an integral part of ensuring business competitiveness. The process of providing a variety of services in the global business environment requires a high level of computer-aided workflows that underlie the various services, due to the constant change of requirements for them. Besides, workflows that perform an ordered sequence of computations in a virtual digital space during providing services to the end user are formed based on algorithms for executing an ordered sequence of domain actions.

The sequence of actions (tasks), ordered in a logical and orderly way with cause and effect relationships between them, the purpose of which is to obtain a result (product, service, information, decision, etc.) is called a *business process*.

Workflow is a software tool (complex software component) used to execute a business process of a domain in a virtual digital space, which is both a graphical and a code representation of the execution sequence of software components (services) in a computing environment, including specific calculations, information dependencies and sequences of decisions and computational works. Thus, it is a set of technologies and tools that allow you to appropriately follow a certain sequence of work of business processes of the domain, to implement certain computational procedures to achieve the goals.

Computer-aided workflows with the help of modern information technologies is carried out, as a rule, to reduce the number of failures, errors, downtime and losses, increase the reliability and accuracy of procedures. With the implementation of workflow modeling and computer-aided design, it is possible to deeply analyze the situation and develop a plan of action to reduce costs and increase productivity overall.

Workflow Management System is a software system that serves to prepare, execute and monitor a defined workflow order and tasks to improve productivity and reduce the cost required to complete the workflow.

The following elements are required to develop a workflow scenario:

Method: A series of steps for submitting and modeling domain information.

Meta-Model: An information model that describes the properties of the individual components of the domain that are important in terms of content, as well as their meaningful relationships that determine the conditions of using the individual components.

Notation: Symbols and rules for presenting information.

Tools: Computing tools (software) for information processing.

To represent both sequential and state-based workflows, it is suggested to use ontology as their meta-model. In terms of modern software workflow is a sequence of instances of a class (or subclass) in the ontology, representing the individual stages (states) of the computational process.

An important aspect of the need to use ontology for workflows generation is the distribution, heterogeneity, and multiagency of components of the global virtual (digital) business environment. For example, ontologies were developed for the automotive industry, describing the entire logistics chain of production (Odette [2], RosettaNet [3]).

Due to the transition to digital space and the constant modification of business processes in all industries under Industry 4.0, as well as the implementation of artificial intelligence systems, the need to develop specific ontologies is expected to increase. Accordingly, there is a growing need for workflow automation methods based on the computer-aided generation of their steps from ontological models of both the domain and the global virtual business environment. According to [4], research in this area is at an early stage.

Taking into account that the ontology of workflows allows you to formally describe the process, structure and visualize parts of it, combining functional services with links in the form of domain constraints that reflect the interaction between them, and the ontology of functional services will retain their formal meta descriptions, an intellectual environment based on which it is possible to computer-aided workflows generation depending on the data state of the domain.

An advantage of the ontological approach is that the ontological model presents the relationships between computational components and the rules for their application in the form of graphs, allowing the use of graph theory mathematical apparatus to generate workflows in real time, based on the description of the input and output parameters of each functional service.

This paper is organized as follows. After the Introduction section 2 contains a state of the art approach to computer-aided workflow design, methods and tools for their analysis. Section 3 is dedicated to the formalized description of ontology-based meta-models. Section 4 provides an example of software implementation and further research perspectives. Section 5 includes a summary and outlook on future work.

II. State of the Art and Background

Many studies have been devoted to the computer-aided workflow generation [2-18], by which computational independent workflows are developed using graphical standards, allowing them to be formalized to reflect possible flows and transitions in a schematic form. The analysis showed that in practice, computational independent workflows are usually developed using graphical notation by tools such as BPMN 2.0 [5], UML AD, USLD [6] and tools such as CA ERwin Process Modeler [7], Enterprise Architect [8] and MS Visio [9].

A brief overview of workflow analysis methods and tools showed that there are two types of analysis that take into account the performance of the computing process:

1.Analysis of design time (modeling and verification). Monte Carlo modeling tools and Petri analysis can be used. For example, there are analysis tools for BPMN, UML AD [12], EPC [13], BPEL [14] using the Petri net mathematical apparatus and their subsequent analysis. USLD diagrams can be analyzed using ontological analysis of services [15].

2.Analysis of execution time (e.g., process of determining execution time based on execution logs) [16].

Software such as Pegasus, Cactus, ASKALON, GLUE, etc. is used for such areas of analysis [17]. All mentioned and analyzed current opportunities for this stage of the task are very limited. The disadvantages of methods and tools of workflow analysis are clearly described in [18]. The central disadvantage is that the requirements analysis stage is mainly applied manually.

Thus, it can be concluded that the strategies of adapting systems to constantly changing user requirements through computer-aided design and re-engineering of end-user services (i.e. designing and modifying existing business processes) are poorly formalized and verified, requiring several iterations with the participation of analysts, system architects, significant investment of time and resources.

There are quite a few tools for describing and building workflows, but all of them are poorly automated and require human involvement. There are currently no solutions that would allow you to reengineer workflows due to changing meta descriptions of functional services.

III. Formalized Description of Workflow Generation from Ontology

In general, a workflow is a scheme of organizing a sequence of execution tasks within a single service that is provided to end users, which may include related computational subprocesses, information dependencies, decision making sequences, and some computations.

A flowchart or graph consisting of operations (functional services or microservices), symbols of logic, connections is used to represent the workflow. The sequence of calculations is indicated by arrows.

Typically, the workflow can be described using formal or informal flowchart methods that show directional flows between processing steps. Individual processing steps or workflow components can be basically defined by three parameters:

- Input description: information, data, and resources required for the workflow.
- Conversion rules: algorithms that can be executed by a person, program, or both.
- Output description: information, data and resources that are processed by the workflow and ready to be forwarded.

Components can be combined only if the output information of one component corresponds to the input information of the component, which performs further processing, and satisfies the conditions of the domain. Thus, a semantic description must include a description of the input, output and meta description of the functional services performed in this workflow. If there are several methods (functional services) for processing the same data, then you need to add a meta description of the algorithms that perform the given function, and it is desirable to specify the performance characteristics of the given functions, such as accuracy, speed and others.

This is due to the fact that the information that the workflow is weakly cohesive, multi-structured and can be processed by different methods depending on the characteristics of the input flow. But from a data processing point of view, information needs to be structured and the system trained on specific data in order to be able to use different methods more effectively.

A. Description of the Service Presentation Model of the Workflow Generation:

In many cases, the same problem can be solved by different methods. The nature of data processing is determined by the probabilistic properties of observations made on some data, so it is proposed to use ontological models to describe all the rules for establishing connections based on input data in order to determine the most effective method of solving the same problem.

Consider a computer-aided workflow generation approach based on the domain ontologies, computing environment, namely workflows and all the components (functional services, relationships, and rules) that form the workflow.

Functional Service (S) is a service, an element (object) that accepts a dataset, performs operations on it, and transmits new data to the output. Functional services are saved using a file system, and service meta descriptions are in the database. All meta descriptions of the functional services of each domain are stored in separate database collections. Functional services can be simple and complex.

Consider a simple service model:

$$S = \{Code, M, P\},$$

where:

$Code$ - the functional code of the functional service;

M - meta description of the functional service;

P- the physical path to a functional service.

Functional service meta description (M) is a description of a function, its input and output parameters, to be further displayed in the user interface and to take into account constraints when building relationships in the workflow.

A meta description of a simple service can be represented as:

$M = \{ID, F, \{I_1, \ \dots \ I_n\}, \{O_1, \ \dots, \ O_n\}, T\}$,

where:

$M = \{ID, F, \{I_1, \ \dots \ I_n\}, \{O_1, \ \dots, \ O_n\}, T\}$ – a meta description that includes the following parameters:

ID – a meta description identifier, for example: ObjectID («5cdd4fd0598ec622d023e048»)

$F = \{N, D, R, P\}$ – a set of parameters for describing a functional service;

$I_i = \{N, D, R, T\}$ – the set of parameters to describe the input parameters;

$O_i = \{N, D, R, T\}$ – a set of parameters to describe the output parameters, where:

N – the name of the object (functional service, input / output parameters), such as *"Sustainability calculation"*

D – the description of the object, the characteristic of the computing operations of the functional service, for example – «The service checks whether this card exists at the destination bank»;

R – the name of the object that is convenient for a person to understand, for example – «Checking the presence of a payee's bank account»

P – the path to a functional service, for example – «./bank/checkBankAccount.js»;

T – type of service, for example – «function», «process»; variable type «number», «string» ...

$Proc = \{CPU, R, t_{max}, \dots\}$ – a description of non-functional service parameters, where:

CPU – the number of computing kernels required to perform the service;

R – the amount of RAM;

t_{max} – maximum time for service execution;

$\dots$ – other parameters.

The meta descriptions of the functional service are stored in the database as follows:

```
{
 _id
 function: {name, description, readable_neme, path},
 input: [{name, description, readable_neme, type}],
 output: [{name, description, readable_neme, type}],
 proc: {cpu, ram, tmax, ...}
 type
}
```

In algebraic form, a complex service can be written:
$S_{complex} = \{I, \{R_1, \ \ldots, \ R_n\}, \{S_1, \ \ldots, \ S_n\}, O, Proc\}$,
where:
$S_{complex} = \{I, \{R_1, \ \ldots, \ R_n\}, \{M_1, \ \ldots, \ M_n\}, O, \}$ – a set of parameters that describe a complex service, namely:

I – service input;

R_i – the rule of choice of method;

$S_i = \{I \bigcap R_i\}$ – a data processing method or simple service that can be selected when applying the R_i rule to the input parameters;

O – the output of the service.

$Proc = \{CPU, R, t_{max}\}$ – a set of non-functional parameters that acquire the value of the selected processing method.

B. Formalized Workflows Description

The ontology of workflows consists of functional services and relationships between them, which is built taking into account the limitations of the domain.

A domain is a set of all objects, classes, and attributes linked together by logical connections.

Workflow – a process that consists of one or more functional services interconnected by the domain and logical connections.

The workflow is represented as follows:

$$W_f = \{I, \{F_1, \ \ldots, \ F_n\}, O, Proc\},$$

where:

$W_f = \{I, \{F_1, \ \ldots, \ F_n\}, O, Proc\}$ – a workflow that includes the following parameters:

I – description of the input parameters;

F_i – used functional services;

O – description of the output parameters;

$Proc = \{CPU_{max}, R_{max}, t_{max}\}$ – non-functional workflow parameters where:

CPU_{max} – the maximum number of computing kernels required;

R_{max} – the maximum amount of RAM required;

t_{max} – the maximum permissible cumulative runtime that consists of the sum of t_{max} of all functional services.

Relationships between services (L) – logical connections establish the order of data processing, the order of transferring parameters from one service to another, can be established only with the condition that the two services are logically connected and at least one of the output parameters of the first service corresponds to at least one input parameter of another service.

C. Formalized Description of Workflow Support Infrastructure

In computer-aided workflow systems, their workflows are accomplished through workflow management systems, which are described using data flow diagrams.

Formally, the workflow management system can be represented by:

$$WfMS = \{S_r, S_d, S_c, S_a, S_{ai}\},$$

where:

S_r – routing system, routing the flow of information or objects, transmits information from one work item to the next;

S_d – a distribution system that detects exceptional circumstances and transmits information to designated system components;

S_c – coordination system, coordinates simultaneous activities, preventing resource conflicts (allocation of CPU time and memory) or conflicts of priorities;

S_a – agent system, automatically starts the service, controls its execution and receives data from it;

S_{ai} – a helper system that regulates the choices of the following methods and provides suggestions which of the following methods can be used.

The components of a Data Flow Diagram (flowchart):

$$DFD = \{D_p, D_{df}, D_w, D_t\},$$

where:

D_p – the process, part of a system that converts inputs into outputs;

D_{df} – data flow, shows the transfer of information from one system to another;

D_w – a data storage used to store data for later use

D_t – access terminal, interface of the system interaction with the external environment.

D. Ontology-based Computer-aided Workflows Generation

To generate a workflow, you must perform an intersection operation with each one for all services, provided that there is a logical connection, as well as the data connection between them. In this case, the input data will determine the service from which the calculation begins, provided the data coincides with their meta descriptions (M) in the service. The meta descriptions (M) of the services (S) are a set of:

$$M = \{M_{use}, M_{domain}, M_{all}\},$$

Formally, the workflow can be represented as:

$$W_f = \{\forall M_{use,i} \bigcap \forall M_{use,j} | \exists L_{ij}\},$$

where:

$W_f = \{\forall M_{use,i} \bigcap \forall M_{use,j} | \exists L_{ij}\}$ – a generated workflow consisting of:

$M_{use} = \{M_{def} \bigcap M_{domain}\}$ – a selection of services used in the workflow where:

M_{use} – functional services used in the workflow;

M_{def} – user-defined functional services;

M_{domain} – functional services of this domain;

$L_{ij} = \{M_{use,i}.O = M_{use,j}.I\}$ – define workflow relationships based on compatible input and output rules, where:

L_{ij} – workflow connections;

$M_{use,i}.O$ – the output parameter of the i-th functional service;

$M_{use,j}.I$ – the input parameter of the j-th functional service;

$M_{domain} = \{M_{all} \in SD\}$ – selects all domain-specific services where:

M_{domain} – functional services of a domain;

M_{all} – all functional services available in the repository (register);

SD – Specified domain.

The steps of workflow generation using an ontological register of services based on a formal algebraic system are described below.

Step 1 – Determine the parameter/parameters (meta descriptions M) of the service / microservice (S) that needed to solve a specific problem.

Step 2 – Selection the service / microservice parameter values given in the ontology as input. In this case, the parameter values can be *static* (predefined by the expert) and *dynamic* (the user sets the parameter values in the expert's range «from» – «to»).

Step 3 – Definition of microservices and their clustering into services on the basis of simple operations: *elementary* (addition, multiplication), *logical* (conjunction, disjunction, equivalence, negation, implication), *over sets* (union, cross-section, belonging, etc.), according to the specified rules.

Step 4 – Compare the values of service / microservice parameters based on the intersection operation with each for all services / microservices. In this case, based on the static values of the parameters specified in the ontology, a predefined service / microservice is selected from the register, and on the basis of dynamic values, a number of services / microservices are offered to the user by which the problem can be solved in the best way (faster, easier).

Step 5 – Combining multiple services / microservices into a workflow.

The order of inclusion of services in the workflow is determined by the presence of parameter values in stages 2 and 4. For example, the $n+1$ service will be executed after the n service in the workflow if the input for its execution is data obtained from the n service execution.

Changing the order of the steps will result the wrong answer, which makes it impossible to use the inversion operation and the logical combination operation for the selected service.

IV. Example of Software Implementation Of Domain Ontology-based Workflows Generation

The domain «Temporary norms for calculation of strength of power elements of ITER magnetic systems» is chosen as an example. Services are engineering calculations of superconducting electromagnetic systems – choice of basic parameters and calibration calculation, and microservices – separate calculations of selected parameters.

The service ontology was generated by TODOS information technology [19]. It consists of 2 classes, 10 subclasses and 53 terminal nodes (56 nodes in total), linked by «IS-A» type connections and has the format *.xml and contains information about each service / microservice as well as their input parameters. The service selected for the example, «Stability calculation» is described by metadata, the values of which are calculated during the computer-aided workflow. Tools for viewing ontologies in the TODOS system will allow to present the register of services in the form of a frame, hierarchically ordered applications, graph and a table. Searching for services in the register to run an engineering workflow is performed by IT TODOS based on an algorithm for solving a choice task by ranking alternatives by a set of metadata values [20].

Using the XML file parsing / generation modules, the original file was converted to JSON format. Further processing of the ontology file allowed us to generate a model describing the workflow in BPMN format.

The bpmn-js module visualized the workflow, data flow and variables in the web user interface (Fig. 1), which allowed the user to edit the input at each stage of the process.

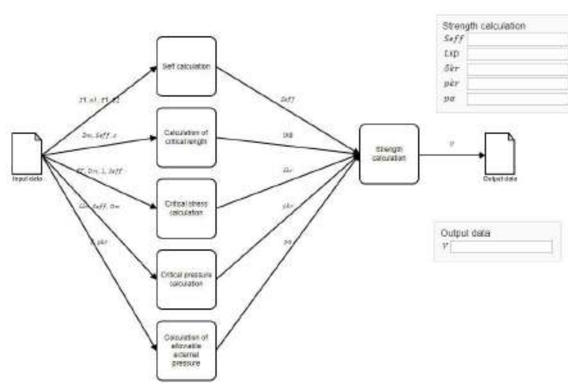

Figure 1. BPMN-visualization of workflow

Each service has its own GUID (Globally Unique Identifier) – unique key, by which each service workflow can be activated from the service ontology.

Software implementation of the computer-aided workflow algorithm involves the formation of a plan for its execution, that is, the sequence of execution of its services by BPEL-PM. Existing modern software tools and environments, in conjunction with the TODOS Ontology Model System, allow the creation and application of ontology models for various domains, form the single distributed intelligent business environment support platform for Industry 4.0.

V. Summary and Outlook on Future Work

The paper proposes an approach to computer-aided workflows design of and their components (functional services, communications and rules of interaction) by computer-aided generation of both a set of services that are components of workflows and sequence of their execution, using ontologies – meta-models of work process, services, domain, logical rules that establish relationships between functional services.

A formalized description of ontological domain models, functional services and workflows, as well as operations of computer-aided workflow generation using relationships and rules of application established between ontological models, are offered.

The proposed approach makes it possible the computer-aided workflows generation from functional services described in the domain ontology, and management the computation process depending on the input data.

The use of the proposed approach to the workflow generation makes it possible to computer-aided the choice of processing method (functional service) of input data, which is a very important factor in real-time systems. Depending on the data, the most efficient method for faster processing can be selected.

Using the domain ontology as a register of meta descriptions of functional services will help computer-aided workflow generation and define a functional service from many similar services that most closely match the conditions of use determined by the input data flow.

Further studies will be devoted to a more detailed consideration of the process of computer-aided workflows generation from sets of functional services, in particular, the computer-aided generation of program code for the execution of workflows.

References

[1] Industry 4.0: Building the Digital Enterprise. Available at: https://www.pwc.com/gx/en/industries/industries-4.0/landingpage/industry-4.0-building-your-digital-enterprise-april-2016.pdf

[2] Odette. Available at: https://www.odette.org/

[3] W. F. Boh, C. Soh, S. Yeo. Standards development and diffusion: a case study of RosettaNet. *Communications of the ACM*, 2007, vol. 50 (12), pp. 57–62.

[4] Flensburg P. Generation of ontologies for workflows. IN FLENSBURG P. & IHLSTRÖM C. (Eds.) *Learn IT, Know IT, Move IT – IRIS27*. Falkenberg, 2004. [Online]. Available: http://www.perflensburg.se/texts/iris27-1118.pdf

[5] Business Process Model And Notation. Available at: http://www.omg.org/spec/BPMN/2.0/.

[6] I. Grobelna, M. Grobelny, M. Adamski. Model Checking of UML Activity Diagrams in Logic Controllers Design. *Proceedings of the Ninth International Conference on Dependability and Complex Systems DepCoS-RELCOMEX, Advances in Intelligent Systems and Computing*, 2014, Vol. 286, pp. 233-242.

[7] ERwin Process Modeler. Available at: https://erwin.com/products/erwin-business-process/ http://erwin.com/products/erwin_process_modeler/.

[8] SPARXsystems. Available at: http://www.sparxsystems.com/products/index.html.

[9] Visio. Available at: http://office.microsoft.com/en-us/visio/.

[10] E. Börger. Approaches to modeling business processes: A critical analysis of BPMN, workflow patterns and YAWL. *Software & Systems Modeling*, 2012, vol. 11. pp. 1-14.

[11] L.Ivancic (2019) Business Process Modeling Vs. Process Mining: A Comparative Study of Methods and Software Tools. *10th Scientific Conference of Croatian Society for Quality*, 2019. pp. 41-41.

[12] Y. Imtiaz Khan, N. Guelfi. Slicing High-level Petri nets. *International Workshop on Petri Nets and Software Engineering (PNSE′14)*, 2014, vol. 1160, pp. 201-220.

[13] F. Kozisek, I. Vrana. Business Process Modeling Languages. *Agris on-line Papers in Economics and Informatics*, 2017, vol. 9, pp. 39-49.

[14] E. Stachtiari, A. Mentis, P. Katsaros. Rigorous Analysis of Service Composability by Embedding WS-BPEL into the BIP Component Framework. *Proceedings – 2012 IEEE 19th International Conference on Web Services*, ICWS 2012, 2012, pp. 319-326.

[15] T. Erçelebi Ayyıldız, A. Koçyiğit. Size and Effort Estimation Based on Problem Domain Measures for Object Oriented Software. *International Journal of Software Engineering and Knowledge Engineering*, 2018, vol. 28, No. 02, pp. 219-238.

[16] L. Silva, R. Braga, F. Campos. Scientific Workflow Composition in E-Science. 25th Brazilian Symposium on Software Engineering, 2011. pp. 273-282.

[17] T. Allard, P. Alvino, L. Shing, A. Wollaber, J. Yuen.. *A dataset to promote automated workflow analysis*. Article (PDF Available) in PLoS ONE 14(2):e0211486, 2019.

[18] T. Kot, A. Reverchuk, L. Globa, A. Schill A novel approach to increasing the efficiency of OSS / BSS workflow planning and design. *BIS 2012 Conference*. Springer, 2012.

[19] V. Yu. Velychko, M. A. Popova, V. V. Prykhodniuk. TODOS – IT platform for the formation of Transdisciplinary Information Environments. *Systems of Arms and Military Equipment*, 2017. Vol. 1, pp. 10-19.

[20] O.Ye. Stryzhak, V.V. Gorborukov, O.V. Franchuk, M.A. Popova. Ontology of the problem of choice and application in the analysis of limnological systems. *Environmental Safety and Natural Resources*, 2014, vol. 15, pp. 172-183.

Генерация рабочих процессов из онтологии предметной области

Глоба Л. С., Попова М. А., Бакай А. Н.

В работе освещён подход к автоматизации проектирования рабочих процессов и их компонентов (функциональных сервисов, связей и правил взаимодействия) путем автоматизированной генерации как набора сервисов, которые являются составляющими рабочих процессов, так и последовательности их выполнения, используя онтологии – мета-модели рабочего процесса, сервисов, предметной области, логических правил, устанавливающих связи между функциональными сервисами.

Применение предложенного подхода позволило генерировать рабочие процессы на основе параметров, определенных в мета-моделях (онтологиях) реестра сервисов, и ограничений, наложенных онтологией предметной области, учитывая как функциональные, так и нефункциональные требования, реализуя преобразования рабочего процесса независимо от вычислений в его программной модели выполнения.

Для автоматизированной генерации рабочих процессов разработана программная среда, работоспособность которой проверена на примере генерации рабочего процесса с применением инструментальных средств построения и модификации онтологий ИТ ТОДОС и генерации схемы рабочего процесса средствами языка BPMN.

Received 12.12.2019