Interoperability Constraints in Service Selection Algorithms

Paweł L. Kaczmarek
Faculty of Electronics, Telecommunications and Informatics, Gdańsk University of Technology,
Narutowicza 11/12 Str., 80-233 Gdańsk, Poland

Keywords: Service Oriented Architecture, Interoperability, Service Integration, Business Process Management, Quality of Service.

Abstract: In Service Oriented Architecture, composite applications are developed by integration of existing, atomic services that may be available in alternative versions realizing the same functionality but having different Quality of Service (QoS) attributes. The development process requires effective service selection algorithms that balance profits and constraints of QoS attributes. Additionally, services operate in a heterogeneous environment, which requires resolution of interoperability issues during integration. In this paper, the author proposes a methodology that introduces interoperability analysis into existing service selection algorithms. Algorithm data structures are extended with additional constraints that represent interoperability for the two considered computational models: the graph-based model and the combinatorial model based on integer linear programming. The extensions enable a straightforward application of a wide range of existing algorithms as the general structure of input data is preserved. As a part of the research, a system that supports development of SOA-based applications was implemented. Chosen service selection algorithms together with appropriate extensions for interoperability analysis were implemented in the system.

1 INTRODUCTION

Service Oriented Architecture (SOA) allows us to reduce development cost and time by extensive reuse of existing service components and competition between service suppliers. Alternative services realizing the same functionality differ in non-functional, Quality of Service (*QoS*) attributes such as performance, reliability and price. Typically, application development intends to optimize the *QoS*, which means maximization of desired attributes, such as performance, while simultaneously meeting constraints imposed on the final application, such as maximum price or minimum reliability (Yu et al., 2007) (Cao et al., 2007a). It is required that services with optimal *QoS* values are selected, while constraints are satisfied.

Existing methods and algorithms of service selection, however, focus on the formal model of the complex service refraining from interoperability issues that may affect the integration process as services are deployed in heterogeneous runtime environments. Technical interoperability limits the choice of alternative services despite theoretical conformance of functionality and correctness of *QoS* analysis. Although services typically use the Web services standards (WS*) to exchange information, there are many threats for effective integration. The runtime platforms that host services support only selected standards and versions from the wide set of WS* standards. Even if runtime platforms are compatible in supported standards, interoperability is not guaranteed because of standard inconsistencies and vendor-specific extensions (Fisher et al., 2006) (Egyedi, 2007). A service selection may prove to be technically infeasible if services prove to be non-interoperable. Service selection methods additionally require mechanisms to represent and guarantee interoperability in the final application.

Considering the problem, the author proposes a methodology that includes interoperability analysis in service selection algorithms. The data structures of service selection algorithms are extended with dedicated constraints that represent interoperability between services. The constraints result from interoperability rates that consider runtime environments that host the services and WS* standards that are used by the services. In the methodology, two computational representations are analyzed: the combinatorial model together with algorithms based on the integer linear programming (ILP) problem and the graph-based model together with algorithms based on single source shortest path selection. The proposed exten-

L. Kaczmarek P..
Interoperability Constraints in Service Selection Algorithms.
DOI: 10.5220/0003980500230032
In *Proceedings of the 7th International Conference on Evaluation of Novel Approaches to Software Engineering* (ENASE-2012), pages 23-32
ISBN: 978-989-8565-13-6

sions enable a straightforward application of a wide range of existing algorithms as the extensions preserve the general structure of data models.

The proposed solution was implemented in a web-based system that supports service selection in development of workflow applications as a representative case of the composite service concept. The system maintains a knowledge base of runtime environments interoperability, which includes a catalog of WS* standards, versions and configuration options. Chosen service selection algorithms together with appropriate extensions for interoperability analysis were implemented in the system.

The rest of the paper is organized as follows. The next section presents and analyses background knowledge used in the methodology. Sect. 3 describes interoperability notation adjusted for service selection purposes. Sect. 4 presents extensions for selection algorithms in the graph-based and combinatorial models. System implementation is presented in Sect. 5. Finally, Sect. 6 presents related work and Sect. 7 concludes the paper.

2 SYSTEM MODEL AND BACKGROUND

Service selection during development of composite services is a mature research discipline that has already developed models and solutions for various application areas. In this research, existing models are extended with issues specific for interoperability analysis, considering that interoperability may limit the choice in concrete situations.

The formal model of a composite service may be applied in concrete application areas such as: development of WS*-based services using the SOA approach, development of cloud-based applications and business process modeling (BPM) and execution using workflow management systems.

2.1 Composite Service Model

Typically, the model of a composite service (*CS*) takes the following assumptions (Yu et al., 2007) (Alrifai et al., 2009) (Cao et al., 2007a):

- A composite service consists of N atomic operations composed using composition structures (e.g.: loops, conditions, forks, joins). Each operation is realized by a service from a service class $(S_1, \ldots, S_N)$, a service class is also called an abstract service.
- For each service class S_i, there exist one or more atomic services (s_{ij}) that realize the same functionality, but differ in non-functional attributes. Each service s_{ij} is associated with a *QoS* vector $q_{ij} = [q_{ij}^1, \ldots, q_{ij}^n]$ of non-functional attributes, e.g.: service price, performance and security.
- There is a defined utility function (F) that is the evaluation criteria of service selection on its *QoS* attributes. The utility function depends on user preferences considering the non-functional attributes. The user intends to minimize attributes such as price and maximize performance and reliability with given weights for each attribute.

The *QoS* value of the final *CS* depends on service selection and the structure of the *CS*. The calculation considers probability of execution in multiple execution paths, loop unfolding and others. Considering that *QoS* values are expressed in different metric types, a form of normalization is performed to calculate the utility function F. Attributes receive positive and negative weights depending on whether they are desired or undesired as *QoS* of the service as described in detail in (Zeng et al., 2003) (Yu et al., 2007).

Additionally, it is required that the overall *QoS* value of the composite service does not exceed a specified limit (e.g. a price constraint), which is known as the *QoS* constraint and defined as:

$$Q_c = [Q_c^1, Q_c^2, \ldots, Q_c^m] \quad (1)$$

Service selection process intends to select services such that

1. The utility function F of the composite service is maximized
2. Q_c constraints are satisfied for the composite service ($Q_{cs}^a \leq Q_c^a, \forall_{Q^a \in Q_c}$)

2.2 Existing Approaches to Service Selection

The service selection problem with *QoS* constraints has exponential computational complexity in the general case (Alrifai et al., 2009) (Bradley et al., 1977). Different solutions have been proposed that either narrow service structure to make it feasible for optimal calculations or apply a heuristic algorithm. Existing solutions, however, do not integrate the concept of interoperability analysis with service selection based on *QoS* attributes.

This work intends to introduce interoperability constraints to existing algorithms without causing the necessity of major changes in algorithm structures. In order to cover a possibly wide range of algorithms, the following features are considered:

Table 1: Selected selection algorithms and their characteristics.

Algorithm	Computational model	Optimization scope and accuracy
BBLP, WS_IP (Yu et al., 2007)	Combinatorial	Global optimal
MCSP (Yu et al., 2007)	Graph	Global optimal
RWSCS_KP (Cao et al., 2007a)	Combinatorial	Global heuristic
WS_HEU, WFlow (Yu et al., 2007)	Combinatorial	Global heuristic
MCSP-K (Yu et al., 2007)	Graph	Global heuristic
SEWSCP (Hong and Hu, 2009)	Combinatorial	Local optimal
ACO4WS (Wang et al., 2001)	Graph	Global heuristic (AI)
PSO (Xia et al., 2009)	Graph	Global heuristic (AI)
DaC (Yu et al., 2006)	Combinatorial	Local optimal
LOSS/GAIN (Sakellariou et al., 2007)	Graph	Global heuristic
GA (Cao et al., 2007b)	Graph	Global heuristic (AI)
FastHeu (Czarnul, 2010)	Graph	Local heuristic (DYN)
DIST_HEU (Alrifai et al., 2009)	Combinatorial	Local optimal
QCDSS(Chun-hua et al., 2009)	Graph	Global heuristic (AI)

- **Computational Model.** Two well-known models are used in service selection: the combinatorial model - 0-1 multidimensional multichoice knapsack problem (MMKP), e.g.: (Hong and Hu, 2009) (Cao et al., 2007a), and the graph model - multiconstraint optimal path selection (MCOP), e.g.: (Cao et al., 2007b) (Wang et al., 2001). The work (Yu et al., 2007) analyzes both approaches.
- **Accuracy.** Optimal algorithms are applicable for small services, e.g. (Yu et al., 2007) proposes an algorithm based on the branch-and-bound linear programming. For large services, heuristic algorithms are proposed, such as the shortest-path based MCSP-K algorithm (Yu et al., 2007) or the combinatorial-based RWSCS_RP algorithm (Cao et al., 2007a). Artificial intelligence (AI) heuristic methods are also used, e.g. the ant-colony based ACO4WS algorithm (Wang et al., 2001) and particle swarm optimization (Xia et al., 2009) (Chun-hua et al., 2009). Some solutions propose dynamic (DYN) optimization (Czarnul, 2010).
- **Optimization Scope.** The reduction of optimization scope is another method of solving the problem. In this approach, the composite service is divided into subservices that are analyzed locally, typically using an optimal algorithm, e.g.: SEWSCP (Hong and Hu, 2009), DIST_HEU (Alrifai et al., 2009) and DaC (Yu et al., 2006).

Table 1 shows selected algorithms and their characteristics.

In the MMKP/combinatorial model (Cao et al., 2007a) (Yu et al., 2007) (Martello and Toth, 1987), service classes are mapped to item groups, while atomic services are mapped to items in the groups. Each atomic service (s_{ij}) requires resources q_{ij} and supplies profit F_{ij}. The Q_c constraints are treated as available resource in the knapsack. The problem is formulated formally as:

$$\begin{aligned} & Max \sum_{i=1}^{N} \sum_{j \in S_i} F_{ij} x_{ij} \\ & subject\ to \sum_{i=1}^{N} \sum_{j \in S_i} q_{ij}^a * x_{ij} \leq Q_c^a \ (a = 1, \ldots m) \\ & \sum_{j \in S_i} x_{ij} = 1 \\ & x_{ij} \in \{0,1\} \ i = 1, \ldots, N, \ j \in S_i \end{aligned} \quad (2)$$

where $x_{ij} = 1$ if service s_{ij} is selected for the solution and $x_{ij} = 0$ otherwise. The problem may be solved using integer linear programming (ILP) methods (Bradley et al., 1977).

In the graph-based model (Yu et al., 2007), every atomic service is a node in the graph and potential transitions between services are considered as links. If a service class S_i sends data to a service class S_k then there is a link between every service from S_i to every service from S_j. The node that represents a service s is assigned with QoS attributes of the service and consequently with service utility function F. QoS attributes of a service are added to every link incoming to the node that represents the service. The multiconstraint optimal path problem intends to find a path such that the utility function F is maximized while meeting the Q_c constraints.

If QoS constraints are removed from the selection problem, a simplified model is created, in which services are selected considering solely the utility function F. In this case, local optimization for each service class S_i may be used to select the service that supplies the highest utility value. Despite the simplification, interoperability needs to be considered in this case too, as the communication model of the final composite service remains unchanged.

2.3 Interoperability in Service Integration

Service integration in SOA requires resolution of interoperability issues. Web services standards (WS*) (Booth et al., 2004) were proposed and accepted as the main solution for achieving the task. Currently, there exist approximately fifty standards in different versions apart from the basic SOAP and WSDL. Standards constitute the WS* stack (WS-I, 2004), in which standards for an extended functionality rely on lower level ones. Main functional groups include the areas of: addressing (WS-Addressing), security (e.g. WS-Security), reliable messaging (e.g WS-Reliablility), transactions (e.g WS-Coordination, WS-AtomicTransactions) and others. Business process modeling and execution standards (e.g. Business Process Execution Language (Oasis, 2007)) leverage existing WS* standards assuming that lower layer standards may be used for communication with atomic services. Despite the general success of WS*, effective service integration using WS* is still a challenging task (Egyedi, 2007) (Fisher et al., 2006).

General purpose interoperability metrics have been proposed as universal means of interoperability rating. Typically, the metrics specify interoperability levels that are associated with a required scope of data exchange (Ford et al., 2007). For example, Layers of Coalition Interoperability (LCI) define nine layers: Physical, Protocol, Data/Object Model, Information, Knowledge/Awareness, Aligned Procedures, Aligned Operations, Harmonized/Strategy Doctrines, and Political Objectives. The LISI (Levels of Information Systems Interoperability) metric defines five levels: Isolated (manual), Connected (peer-to-peer), Functional (distributed), Domain (integrated), Enterprise (universal). Sect. 6 presents more detailed discussion concerning existing solutions and methods.

3 INTEROPERABILITY NOTATION FOR SERVICE SELECTION

In our work, interoperability is analyzed on the communication protocol level, which corresponds to Level 1 (Connected) in the LISI metric or Level 2 (Data/Object) in the LCI metric (Ford et al., 2007). We assume that if services are interoperable on that level, it is possible to exchange data between them in the context of CS construction.

For the purpose of this work, we make the following assumptions:

- There exist different runtime environments (denoted $R = \{r_1,...,r_P\}$ that may host services. Each service s is hosted in a runtime environment $r(s) \in R$ that is unambiguously determined by the atomic service s.
- Each runtime environment supports a known subset of existing WS* standards and their versions.
- Each service s requires some WS* standards (in specified versions) to operate correctly. The standards are supported by $r(s)$.

The number of runtime environments is limited, considering that they are application servers supplied by leading industrial companies or communities. Examples of runtime environments include: IBM WebSphere AS, Microsoft IIS, Apache Tomcat/Axis and Oracle Glassfish.

If a service s communicates remotely with a service s' than it is required that s' can interoperate with s. We rate interoperability binary as either interoperable (if data can be transfered correctly) or non-interoperable (if data can not be transfered). Assuming that s sends data to s', let $q^{comm}(s,s')$ denote interoperability modeled as an attribute of communication cost between s and s', defined as follows:

$$\begin{aligned} q^{comm}(s,s') &= 0 \; if \; s,s' \; can \; interoperate \\ q^{comm}(s,s') &= 1 \; if \; s,s' \; can \; not \; interoperate \end{aligned} \tag{3}$$

In this notation, interoperability cost 0 denotes that services are interoperable while interoperability cost 1 denotes that they are not interoperable.

The q^{comm} value is given a priori and depends on conditions encountered in the real environments that host the services. Information about service interoperability is taken from two complementary sources:

- Experimental verification. A test connection is established between s and $r(s')$ to check if data can be transfered correctly within required standards.
- Theoretical calculation. The calculation compares standards required by s and supported by $r(s')$. r' must support standards required by s if s and s' are to be interoperable. The method is easier to apply but less reliable than the first one, as declared standard compliance does not inferr compatibility (Egyedi, 2007).

The flow of data between services depends on the applied communication model. Two well-known models are considered (Tanenbaum and van Steen, 2002) as shown in Fig. 1:

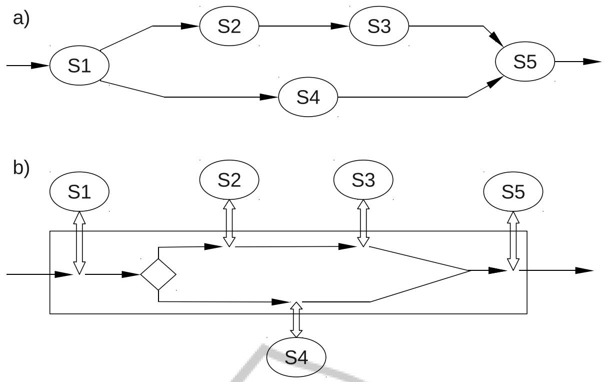

Figure 1: Differences between message passing model (a) and remote method invocation model (b).

- remote method invocation model (RMI)
- message-passing model (MPM)

In the RMI model, there is a central runtime environment that hosts the composite service, denoted ($r(CS)$), and the runtime manages data exchange during invocation of remote atomic services. The model is typically used in the BPM technology by Business Process Execution Language (BPEL) (Oasis, 2007) and Windows Workflow Foundation (the XPDL format). In the message-passing model, services send data directly from one to another without a central coordination point. The Business Process Modeling Notation standard (OMG, 2009) uses this model to describe the logic of a business process.

4 ALGORITHM EXTENSIONS

The proposed extensions of existing models with interoperability constraints enable a possibly straightforward application of existing selection algorithms. We consider both RMI and MPM, as well as graph-based and combinatorial-based algorithms.

4.1 Extensions for Remote Method Invocation Model

In the RMI model, we assume that CS may be hosted on one of alternative runtime environments. Let $R^{cs} = \{r_1^{cs}, ..., r_T^{cs}\}$ denote the set of alternative environments for CS. Each service runtime environment $r(s)$ must interoperate with the chosen runtime environment for the CS. The presented algorithm is applicable if a workflow is defined using the BPEL or XPDL standards.

Let $q^{comm,cs}(s, cs, r^{cs})$ be defined analogously to q^{comm} that is

$$\begin{aligned} q^{comm,cs}(s, cs, r^{cs}) &= 0\ if\ s\ can\ interoperate\ with \\ &\quad cs\ hosted\ on\ r^{cs} \\ q^{comm,cs}(s, cs, r^{cs}) &= 1\ if\ s\ can\ not\ interoperate\ with \\ &\quad cs\ hosted\ or\ r^{cs} \end{aligned} \tag{4}$$

Let $Q^{comm,cs}$ be the set of interoperability information of $q^{comm,cs}(s_{ij}, cs, r_t^{cs})$ for each s_{ij} $i = 1..N$, $j \in S_i$, $r_t^{cs} \in R^{cs}$.

The processing is done as shown in Algorithm 1. The algorithm analyzes in a loop each alternative runtime environment of CS. For each r^{cs}, it performs a pre-filtering of services that are interoperable with the r^{cs}. Then, one optimal solution is calculated for each r^{cs} using an algorithm (SEL_ALG) chosen from existing selection algorithms. Both combinatorial-based and graph-based selection algorithms are applicable in this invocation model. After processing of all r^{cs}, the optimal solution for the whole set R^{cs} is selected

Interoperability constraints introduce a relatively low computational complexity in the RMI model. Selection of interoperable services depends on $O(sizeof(S) * sizeof(R^{sc}))$. In practice, the size of $R^{sc} = T$ may be considered as constant because it does not depend on CS size and there is a limited number of industrially-running environments that may host services. Generation of $tmpSelection$ takes $T * O(SEL\_ALG)$, which does not exceed the computational complexity of SEL_ALG. The final step (selection of the best solution from R^{sc}) takes less than $T * timeof(F)$, which is certainly lower that SEL_ALG, as the algorithms invoke F for each service and for the whole workflow. Therefore, $O(SEL\_ALG)$ is the factor that determines the final computational complexity of Algorithm 1.

Algorithm 1: Service selection using an existing selection algorithm and interoperability constraints in the RMI communication model.

input: $R^{cs}, S_1, ...S_N$
input: $Q^{comm,cs}$, F - utility function
input: SEL_ALG - an existing selection algorithm
output: $[s_{1a_1}, ..., s_{Na_N}]$, $a_i \in S_i$ //optimal selection
  $tmpSelection = \varnothing$ //list of optimal selections for each $r^{cs} \in R^{cs}$
  for all ($r_t^{cs} \in R^{cs}$) **do**
    for all (S_i, $i = 1, ..., N$) **do**
      $S_i^{r_t^{cs}}$ = select all s_{ij} from S_i such that $q^{comm,cs}(s_{ij}, r_t^{cs}) = 0$
      if ($S_i^{r_t^{cs}} == \varnothing$) **then**
        //there are no services in S_i that can
        //interoperate with r_t^{cs}, no feasible
        //solution for runtime environment r_t^{cs}
        continue with next r_t^{cs}
      end if
    end for
    $tmpSelection += SEL\_ALG$ (($S_i^{r_t^{cs}}$, $i = 1..N$))
  end for
  $maxTmpUtility = 0$
  for all ($sel \in tmpSelection$) **do**
    $tmpUtility = F(sel)$
    if ($tmpUtility > maxTmpUtility$) **then**
      $maxTmpUtility = tmpUtility$
      $result = sel$
    end if
  end for
  return $result$

4.2 Extensions for Message Passing, Combinatorial Model

In this model, if a service s communicates with a service s', then $r(s)$ must interoperate with $r(s')$. The model is applicable for workflows defined using the BPMN standard. As stated in Sect. 3 $q^{comm}(s_{ij}, s_{kl})$ denotes interoperability between s_{ij}, s_{kl}, $s_{ij} \in S_i$, $s_{kl} \in S_k$, $i, k = 1, ..., N$, $j \in S_i$, $l \in S_k$. We extend the existing combinatorial model and its corresponding integer linear programming representation with additional constraint equations that represent interoperability:

$$\sum_{j \in S_i} \sum_{l \in S_k} q^{comm}_{s_{ij}, s_{kl}} * (x_{ij} * x_{kl}) = 0 \tag{5}$$

for all pairs S_i, S_k such that S_i sends data to S_k in the structure of CS. It is required that if two services are selected ($x_{ij} = 1$ and $x_{kl} = 1$), then their interoperability communication cost $q^{comm}_{s_{ij}, s_{kl}} = 0$. If two service classes do not pass data, the q^{comm} value is insignificant and may be ignored.

The number of added equations is equivalent to the number of edges in the composite service graph, which depends linearly on the number of services (N). The maximum number of elements in each equation is $Max(sizeof(S_i)) * Max(sizeof(S_i))$ for $i = 1...N$. We assume that the maximum number of alternative services for one service class does not exceed a constant ($MaxS_i$).

Equation 5, however, introduces non-linearity, so the problem is changed from ILP to mixed integer nonlinear programming (MINLP), which makes it significantly more difficult to solve. We transform the equation into a separable programming problem, which is a special class of MINLP that is relatively easily solvable by existing methods (Bradley et al., 1977). In order to achieve the separable programming form, we perform the following transformation:

$$\begin{aligned} x_{ij} * x_{kl} &= y^2_{1,ijkl} - y^2_{2,ijkl} \\ where\ \ y^2_{1,ijkl} &= 0.5(x_{ij} + x_{kl}) \\ y^2_{2,ijkl} &= 0.5(x_{ij} - x_{kl}) \end{aligned} \tag{6}$$

for each $j \in S_i$, $l \in S_k$. After the transformation, additional y variables are added to problem formulation. The number of y variables equals the total number of elements in Equation 5 and has the complexity of $O(N * MaxS_i * MaxS_i)$. The newly created model is more complex than the original one, but it remains solvable by existing methods such as branch-and-bound or implicit enumeration (Bradley et al., 1977). The model is applicable for service selection methods that leverage (N)LP to find an optimal solution, that is for methods based on local optimization (DaC (Yu et al., 2006)) or applied to small composite services.

4.3 Extensions for Message Passing, Graph-based Model

In this model, we extend the composite service graph with additional information regarding interoperability constraints. Analogously to the combinatorial approach, the model is applicable for workflows defined using the BPMN standard. We use the previously defined q^{comm} attribute to represent interoperability communication cost. The processing is done as follows:

- Assuming that a service class S_i communicates with a service class S_k, the $q^{comm}(s_{ij}, s_{kl})$ attribute is assigned to each link from a service $s_{ij} \in S_i$ to a service $s_{kl} \in S_k$. The attribute has the value 0 if services are interoperable and the value 1 in the other case.

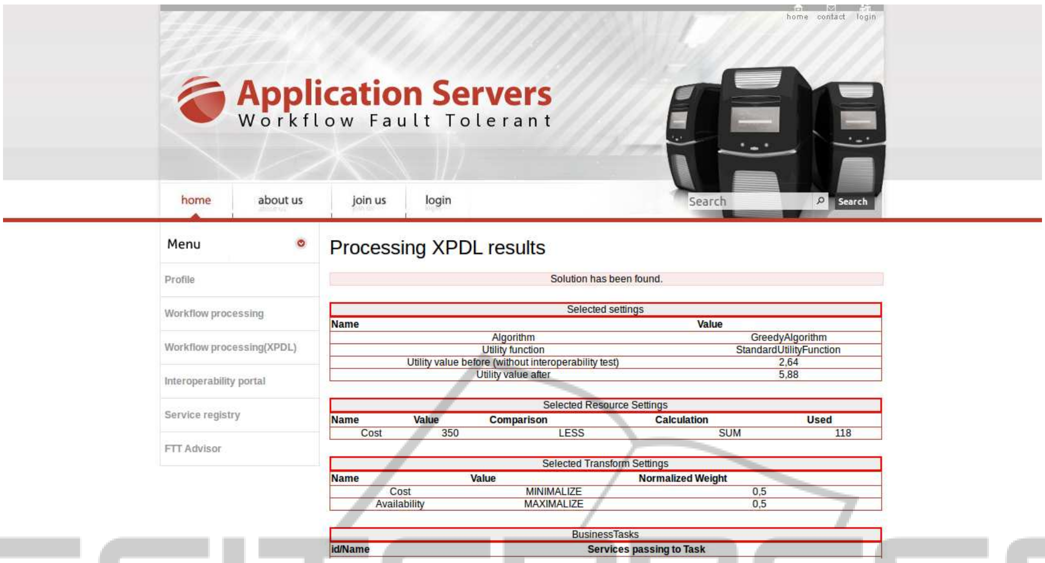

Figure 2: A screenshot of the WorkflowIntegrator system with summary results of an exemplary service selection.

- The Q_c vector is extended with an additional element Q_c^{comm} that represents interoperability, that is:

$$Q_c' = [Q_c^1, Q_c^2, \ldots, Q_c^m, Q_c^{comm}] \\ Q_c^{comm} = 0 \tag{7}$$

  where Q_c^{comm} for a service selection (a *path* in the graph) is calculated as follows:

$$Q_c^{comm}(path) = \sum_{link \in path} q_{link}^{comm} \tag{8}$$

  that is: every edge on the selected path must have the communication cost 0.

The proposed model does not change the graph structure and the structure of Q_c constraints. Therefore, it is possible to apply both optimal and heuristic existing algorithms that are based on graph structure processing, such as MSCP, MSCP_K or ACO4WS (Yu et al., 2007) (Wang et al., 2001).

In this model, interoperability analysis requires much lower computational cost as compared with the combinatorial approach. One additional attribute is added to the *QoS* vector on the service level and on the constraints level. This requires additional computations during graph preparation and during service selection, but does not change the computational complexity of algorithms. Therefore, the graph-based algorithms are more adequate for interoperability analysis in the message passing model.

5 SYSTEM IMPLEMENTATION

As a part of the research, we implemented the WorkflowIntegrator system that enables selection of services during composite service development. The aim of the system is to support developers in the selection process depending on given requirements on utility function, *QoS* attributes, and interoperability constraints. The system is available online at:

http://kask.eti.pg.gda.pl/WorkflowFTI

Fig. 2 shows a screenshot of the WorkflowIntegrator system with summary results of an exemplary service selection using cost and availability as selection attributes.

We use workflow applications as a concrete implementation of the composite service concept, which enables us to leverage existing solutions in service description and invocation.

In the workflow methodology, a workflow application corresponds to a complex services, while a service available through Web services corresponds to an atomic service. Concrete runtime environments are Workflow Management Systems (WfMS) for the workflow application, and Application Servers for atomic services. Three main standards are used for service description: Business Process Execution Language (BPEL), Business Process Modeling Notation (BPMN) and the Windows Workflow Foundation (XPDL format).

The system supplies the ASInteroperability module that stores general purpose interoperability ratings of runtime environments (Kaczmarek and Nowakowski, 2011) available also at:

http://kask.eti.pg.gda.pl/ASInteroperability

The ASInteroperability module contains a systematic description of existing WS* standards, their versions, configuration options and acceptable values for the options. For example WS-AtomicTransaction anticipate the Commit Protocol option with values: Completion, Two-Phase Commit (Volataile), Two-Phase Commit (Durable). Popular existing runtime environments have been described in the system including: name, vendor, version and used libraries. The runtime description model enables dynamic association between runtime environments and dependant libraries.

Interoperability rates are given to concrete configurations in which integration is attempted. Each configuration covers specification of: the two integrated runtime environments, used WS* standards, standard versions, significant configuration options and values settled for the options. Rates are given by developers who have already attempted to establish a concrete integration and share the knowledge with the community. The developer that gives a rate specifies the configuration, the interoperability rate and the required development complexity to establish the integration. The complexity ranges from simple GUI operations, through modification of configuration files, to changes of undocumented features, and indicates what is the required expertese level to establish the integration.

Additionally, a service registry is defined that contains specification of services together with their location, QoS attributes, the runtime environment that hosts the service, and used WS* standards. Additionally, it is possible to specify groups of alternative services, which is the main function of the registry in the context of service selection during service composition. Groups of alternative services are configured manually by developers of the composite service.

Using ASInteroperability and Service Registry, the module of Workflow Fault Tolerant Integrator (WorkflowFTI) supports design of optimal workflow applications depending on the following information acquired from the developer:

1. The developer defines groups of alternative services that correspond to service classes in the *CS* model. Each service contains description of *QoS* attributes, used WS* standards and the application server that hosts the service.
2. The developer specifies the utility function that should be optimized in the form of weights for *QoS* attributes and the method for attribute aggregation (average, sum, max, min and others).
3. The developer specifies which selection algorithm should be used.
4. The user supplies a defined workflow application together with initially assigned services for tasks. The assignment aims at specifying which service group should be used for processing each task.

After the configuration, WorkflowFTI uses the chosen algorithm to select appropriate services and returns the result. During the selection, it considers the relevant information such as: interoperability constraints, *QoS* attributes, utility function and others.

The system contains prototypical implementations of representative existing algorithms extending them with relevant interoperability processing. Both RMI and message passing models are analyzed during the work. The BPEL standard was used as a representation of the RMI model, while the BPMN standard (using the XPDL format) was used as a representation of the message passing model. In the message passing model, interoperability between runtime environments of individual services is analyzed during the selection process. The proposed methodology was implemented using both optimal and approximate existing algorithms. Optimal algorithms include: explicit enumeration based on ILP and a simplified version of the MSCP algorithm. Additionally, we implemented a greedy algorithm as an example of the approximate approach. In the RMI model, services are pre-filtered considering their interoperability with each WfMS, and services supplying the best *QoS* are selected. We verified interoperability analysis using an optimal algorithm based on explicit enumeration, and an approximate algorithm based on local optimization.

The system is implemented in the .NET 3.5 environment and hosted on Microsoft Internet Information Service. We used the MSSQL database for storing data regarding interoperability description and service registry. A web interface is supplied for users intending to design a workflow application or verify interoperability between application servers. Internally, the system uses Web services to integrate AS-Interoperability, Service Registry and WorkflowFTI.

6 RELATED WORK

Sect. 2 presented core concepts regarding the problem addressed in this paper. Apart from the background, various detailed solutions have been proposed. (Fang et al., 2004) presents an interoperability assessment model for service composition using five interoperability levels: signature, protocol, semantic, quality and context. The assessment is based on a formal evaluation metric that assigns a rate in the range $[0,1]$ for each level depending on defined conditions. The work (Bhuta and Boehm, 2007) addresses a similar issue assuming that components-off-the-shelf are integrated. The work specifies groups of attributes that are used for interoperability analysis, for example: control input technique, data communication protocols, encapsulation of behavior and others. Using the model, authors of that paper propose an interoperability assessment tool. Both solutions do not analyze *QoS* attributes and constraints, such as price, performance, and reliability in service composition.

The Web Services Interoperability (WS-I) Organization (WS-I, 2004) has been established to refine existing standards and promote integration of heterogeneous environments. The organization issues WS-Profiles that impose additional constraints on communication formats, which enables higher interoperability. (Lampathaki et al., 2009) overviews XML-based data transfer standards addressed for B2B systems. The formats concern different layers of application development as compared to Web services, focusing on integration of business-oriented data rather than protocol compatibility. (van der Aalst et al., 2003) surveys standards and solutions related to business process modeling and execution. The work discusses correlation between standards related to business process modeling and formal methods of representation. (Czarnul, 2010) presents the BeesyCluster execution environment that enables definition and execution of business processes with service replacement and dynamic optimization.

The concept of Functional Quality of Service (FQoS) is introduced in (Jeong et al., 2009). FQoS describes functional attributes of Web services using mathematical methods to manipulate and quantify those attributes. Authors analyze FQoS on three levels: term similarity, tree similarity and text similarity. (Tan et al., 2009) proposes to use Petri nets for compatibility analysis during composition of BPEL applications. Authors present a detailed association between Petri net concepts and BPEL control constructs and define a formal procedure of Web services composition. The papers focus mainly on service compatibility analysis rather than selection based on non-functional *QoS* attributes.

(Ullberg et al., 2008) and (Tsalgatidou et al., 2008) present frameworks that enable interoperability rating and analysis in Web services-based systems. Both papers propose service metamodels that cover core service description concepts applicable on the enterprise or on the P2P level. Using the metamodels, interoperability of services is rated in the considered scope. (Fu et al., 2005) is another work that rates interoperability in Web services-based solutions. The work proposes a formal description model that covers peers and operations, such as send, receive and transition. Results achieved using the proposed methods may be beneficial in our work as an additional source of information about service interoperability.

7 CONCLUSIONS

The presented methodology enables application of existing service selection algorithms in cases where interoperability issues must be considered. The remote method invocation model suits best to the proposed solution allowing us to use any of the existing algorithms with minor development and computational effort. The message passing model favors graph-based selection algorithms (both optimal and heuristic) allowing us to preserve the general structure of algorithm data model. The application of the methodology in the message passing model for combinatorial algorithms faces difficulties because of non-linearity of constraint equations. The problem is solvable by existing methods, but requires additional computational effort. As a part of the research, implementation work was performed for a representative set of composite service models and algorithm approaches, which verified the presented approach.

Extension of heuristic algorithms based on the combinatorial model is an interesting area of future work, considering that the algorithms may use proprietary data processing that does not rely directly on the defined ILP representation of the problem. Application of the methodology in industrial projects is another area of future work. The current work covered implementation of the methodology in representative algorithms, but the use of it in industrial projects may give further guidelines for extensions and adjustments. The projects may further supply interesting statistical data, such as the ratio of interoperable services, market share of runtime environments, and differences in *QoS* attributes of services, which will be beneficial in further improvements of service selection methods.

ACKNOWLEDGEMENTS

The work was supported in part by the Polish Ministry of Science and Higher Education under research projects N N519 172337. The author would like to thank students of the Faculty of ETI for implementation and testing work regarding the system for service selection.

REFERENCES

Alrifai, M., Risse, T., Dolog, P., and Nejdl, W. (2009). Service-oriented computing — icsoc 2008 workshops. chapter A Scalable Approach for QoS-Based Web Service Selection. Springer-Verlag, Berlin, Heidelberg.

Bhuta, J. and Boehm, B. (2007). Attribute-based cots product interoperability assessment. In *Sixth International IEEE Conference on Commercial-off-the-Shelf (COTS)-Based Software Systems*.

Booth, D., Haas, H., McCabe, F., Newcomer, E., Champion, M., Ferris, C., and Orchard, D. (2004). *Web Services Architecture, Working Group Note*. W3C.

Bradley, S. P., Hax, A. C., and Magnati, T. L. (1977). *Applied Mathematical Programming*. Addison-Wesley.

Cao, H., Feng, X., Sun, Y., Zhang, Z., and Wu, Q. (2007a). A service selection model with multiple qos constraints on the mmkp. In *IFIP International Conference on Network and Parallel Computing*.

Cao, L., Li, M., and Cao, J. (2007b). Using genetic algorithm to implement cost-driven web service selection. *Multiagent and Grid Systems - An International Journal*, 3.

Chun-hua, H., Xiao-hong, C., and Xi-ming, L. (2009). Dynamic services selection algorithm in web services composition supporting cross-enterprises collaboration. *Cent. South Univ. Technol.*

Czarnul, P. (2010). Modeling, run-time optimization and execution ofdistributed workflow applications in the jee-based beesycluster environment. *The Journal of Supercomputing*, pages 1–26.

Egyedi, T. M. (2007). Standard-compliant, but incompatible?! *Computer Standards & Interfaces*, 29(6):605–613.

Fang, J., Hu, S., and Han, Y. (2004). A service interoperability assessment model for service composition. *IEEE International Conference on Services Computing*, 0:153–158.

Fisher, M., Lai, R., Sharma, S., and Moroney, L. (2006). *Java EE and .NET Interoperability: Integration Strategies, Patterns, and Best Practices*. FT Prentice Hall.

Ford, T., Colombi, J., Graham, S., and Jacques, D. (2007). A survey on interoperability measurement. In *12th International Command and Control Research and Technology Symposium (ICCRTS) Adapting C2 to the 21st Century*.

Fu, X., Bultan, T., and Su, J. (2005). Synchronizability of conversations among web services. *IEEE Transactions on Software Engineering*, 31(12):1042–1055.

Hong, L. and Hu, J. (2009). A multi-dimension qos based local service selection model for service composition. *JOURNAL OF NETWORKS*, 4(5).

Jeong, B., Cho, H., and Lee, C. (2009). On the functional quality of service (fqos) to discover and compose interoperable web services. *Expert Systems with Applications*, 36(3).

Kaczmarek, P. L. and Nowakowski, M. (2011). A developer's view of application servers interoperability. *9th Intr. Conf. on Parallel Processing and Applied Mathematics, Part II, LNCS 7204 (in print)*.

Lampathaki, F., Mouzakitis, S., Gionis, G., Charalabidis, Y., and Askounis, D. (2009). Business to business interoperability: A current review of xml data integration standards. *Computer Standards & Interfaces*.

Martello, S. and Toth, P. (1987). Algorithms for knapsack problems. *Annals of Discrete Mathematics*.

Oasis (2007). *Web Services Business Process Execution Language Version 2.0*.

OMG (2009). *Business Process Model and Notation 2.0 Beta 1 Specification*. Object Modeling Group, http://www.omg.org/cgi-bin/doc?dtc/09-08-14.

Sakellariou, R., Zhao, H., Tsiakkouri, E., and Dikaiakos, M. D. (2007). Scheduling workflows with budget constraints. *Integrated Research in GRID Computing, (CoreGRID Integration Workshop 2005, Selected Papers)*.

Tan, W., Fan, Y., and Zhou, M. (2009). A petri net-based method for compatibility analysis and composition of web services in business process execution language. *IEEE Transactions on Automation Science and Engineering*, 6(1).

Tanenbaum, A. S. and van Steen, M. (2002). *Distributed Systems Principles and Paradigms*. Prentice Hall.

Tsalgatidou, A., Athanasopoulos, G., and Pantazoglou, M. (2008). Interoperability among heterogeneous services: The case of integration of p2p services with web services. *Int. J. Web Service Res.*, 5(4):79–110.

Ullberg, J., Lagerström, R., and Johnson, P. (2008). A framework for service interoperability analysis using enterprise architecture models. In *IEEE SCC (2)*, pages 99–107.

van der Aalst, W., Hofstede, A. H. M. T., and Weske, M. (2003). Business process management: A survey. *1st International Conference on Business Process Management, LNCS*, 2678.

Wang, X.-L., Jing, Z., and zhou Yang, H. (2001). Service selection constraint model and optimization algorithm for web service composition. *Information Technology Journal*.

WS-I (2004). *Interoperability: Ensuring the Success of Web Services*. Web Services Interoperability Consortium.

Xia, H., Chen, Y., Li, Z., Gao, H., and Chen, Y. (2009). Web service selection algorithm based on particle swarm optimization. In *Eighth IEEE International Conference on Dependable, Autonomic and Secure Computing*.

Yu, J., Buyya, R., and Tham, C.-K. (2006). Cost-based scheduling of workflow applications on utility grids. In *IEEE International Conference on e-Science and Grid Computing (e-Science)*.

Yu, T., Zhang, Y., and Lin, K.-J. (2007). Efficient algorithms for web services selection with end-to-end qos constraints. *ACM Transactions on the Web*, 1(1).

Zeng, L., Benatallah, B., Dumas, M., Kalagnanam, J., and Sheng, Q. Z. (2003). Quality driven web services composition. In *Proceedings of the 12th international conference on World Wide Web*, WWW '03.