Journal of Web Engineering, Vol. 16, No.1&2 (2017) 145-160

VERIFYING SOUNDNESS OF GEODATA WEB SERVICE COMPOSITION BASED ON PETRI NETS

NA XU

State Key Laboratory of Coal Resources and Safe Mining, China University of Mining and Technology (Beijing), Beijing, 100083, China

School of Business, Shandong Jianzhu University, Jinan, Shandong 250101, China
xuna1011@gmail.com

SUPING PENG ZHANGANG WANG

State Key Laboratory of Coal Resources and Safe Mining, China University of Mining and Technology (Beijing), Beijing, 100083, China

psp@cumtb.edu.cn millwzg@163.com

Received August 3, 2016

Revised November 16, 2016

The emergence of service-oriented architecture (SOA) has made it possible to establish easily accessible geodata web services and perform distributed geodata processing and modelling, which facilitate the provision of geo information in real time. Composition is an important method for dynamically combining distributed individual services and can be incorporated into geoprocessing workflows. Business Process Execution Language (BPEL) and service specifications provided by the Open Geospatial Consortium (OGC) have become the industrial standards for executing geodata web service composition. However, current geodata web service composition soundness verification is beyond the capabilities of BPEL. Soundness verification in the design process can facilitate efficient and cost-effective geodata web service composition execution. To address this issue, Petri nets were used in this study for geodata web service composition analysis. A geodata web service was modelled based on a service net using Petri nets. The geodata web service composition was modelled based on the composition structure. The soundness properties of the geodata web service composition, such as reachability, boundedness, and deadlock, were also analysed. The proposed approach was shown to provide compliant support for geodata web service composition.

Keywords: web service composition, BPEL, SOA, Petri nets
Communicated by: M. Gaedke & Q. Li

1 Introduction

Over the past few years, with the explosive growth of the Internet and easy access to the Web, geodata applications have evolved from process-oriented, object-oriented, component-oriented, and integrationoriented applications to service-oriented applications. In a service-oriented architecture (SOA) environment—a framework of services and service-based development—geodata applications can be offered as services for basic geodata sharing and interoperability functions. As business requirements in the field of geosciences become increasingly more complex, individual web services become less adequate to satisfy these business requirements. Consequently, various distributed individual web services have to be combined to achieve high-level geoprocessing functions. The set of ordered invocations of web services required to achieve such geoprocessing functions can be considered a process based on SOA. Using specific constraint logic, geodata web services can be combined to accomplish web service composition. Reusing a web service composition can substantially increase the development speed of geosciences applications—a feat that individual services are unable to achieve. In recent years, research on geodata web service composition has focused mainly on technologies for its execution [1]. In addition, existing web service composition specifications mainly support direct composition and execution.

The interoperability and accessibility of geodata web service composition execution have improved geodata web applications in various domains. Standardized interfaces and protocols enable web service composition interoperability, support easy integration of distributed services, and ensure that services are easily accessible.

The Open Geospatial Consortium (OGC) [2] has developed many specifications for geodata services to facilitate interoperability and accessibility of geodata web service composition execution. Service specifications developed include the Web Feature Service (WFS) [3], Web Coverage Service (WCS) [4], Web Mapping Service (WMS) [5], and Web Processing Service (WPS) [6]. WPS specifications can provide standard service interfaces for geodata web service composition, and BPEL can be used for geodata web service composition execution. The WPS specification, in particular, has been widely accepted since its emergence. According to OGC, many software products have contributed to the implementation of the WPS specification [7], such as Deegree [8], PyWPS [9], 521 North [10], and the ZOO Project [11].

In this scenario, the question of how to reuse existing services to reduce the cost of redevelopment has received considerable attention. This has resulted in a series of web service composition specifications being developed, such as the XML Process Definition Language (XPDL) [12], Workflow Language (YAWL) [13], and BPEL [14]. Business Process Execution Language (BPEL) has become the most popular specification in the industry. Although the specification, composition, and execution of geodata web services can be adequately supported by BPEL, as business requirements in the field of geosciences become increasingly more complex, and as a result of execution efficiency requirements, web service composition is increasingly being performed using concurrent activities to increase functionality and execution efficiency. Geodata web service composition soundness properties such as reachability, boundedness, and deadlock cannot be adequately verified in BPEL. Instead, it uses Petri nets to validate the web service composition and makes it possible to discover potential defects during the process design phase. This shortens the defect discovery time, which further reduces the redeployment cost, increases process reliability, and yields better overall results at a lower cost.

Although BPEL cannot verify the soundness of geodata web service composition before execution, soundness verification can facilitate efficient execution of geodata web service composition. However, soundness verification of geodata web service composition is still a challenge. Existing methodologies such as state automata [15], labelled transition systems [16], process algebra [17], and description logic [18] are frequently used as formal tools to model services.

Like these existing methodologies, Petri nets and their extensions [19] are of fundamental interest because they provide modelling approaches for concurrent, parallel, and distributed systems; offer graphical support for the representation and understanding of these basic systems; start from state machines to handle the creation and analysis of models; express the main basic concepts in communication, including waiting and synchronization, independent of any particular implementation language; and provide specifications that are independent of the implementation. Many validation methods have been developed based on various theoretical results and support tools. Ultimately, models based on Petri nets will help us to understand, define, and analyse the behaviour of these systems in the preliminary and early stages of their design. Petri nets can provide not only formalism to depict the internal logic and the message exchange behaviour of such systems but also rich analysis capabilities to support the verification of compatibility and mediation existence of the service composition. Petri nets can be used to analyse geodata web service compositions. In previous research, different specific types of Petri nets for web service composition were summarized. Petri net-based algebra has been proposed to capture the semantics of web service composition and model web service composition. This can be used for the verification of web service composition and the detection of inconsistencies within and among web service compositions [20]. Coloured Petri nets have been proposed to model types of resources managed by web service composition [21]. Time-constrained Petri nets have been proposed to model and analyse time-constrained web service composition [22]. Fuzzy Petri nets can be used to define automatic web service selection based on manual user specifications [23]. BPEL processes can be transformed into Petri nets for process verification [24], detection of web service composition incompatibility, and addition of mediator transitions to correct partial incompatibilities among web service compositions [25]. BPEL or WSCI processes can be transformed into Petri nets to verify reachability, safety, and deadlock [26]. WSADL specifications can be transformed into Petri nets to evaluate the aggregated QoS of web service compositions [27]. An OWL-S definition of web service can be used to generate Petri nets to check the soundness of a web service composition and its replaceability [28]. Timed Petri nets can be defined for representation of web service composition flow based on WSDL specifications [29]. A WS-CDL definition of web service composition can be employed to generate Petri nets for use in simulating timed or prioritized interactions among web service compositions [30]. Automated QoS transactional web service composition selection based on Petri nets has been proposed [31], as have automated QoS web service composition selection based on Petri net coverability [32], automated web service composition selection based on coloured Petri nets [33], automated QoS-transactional web service composition selection based on coloured Petri nets [21], a framework for reliable execution of transactional web service composition based on coloured Petri nets [34], and the use of Petri nets to model web service composition, evaluate QoS, verify reachability and deadlock, and control the execution [35].

This paper proposes a Petri net-based approach to modelling and analysis of geodata web service composition. The remainder of this paper is organized as follows. Section 2 outlines the proposed process for modelling geodata web service composition based on Petri nets. Section 3 discusses the analysis process for geodata web service composition. Section 4 summarizes our conclusions and plans for future work in relation to geodata web service composition.

2 Modelling geodata web service composition based on Petri nets

2.1 Geodata web service description

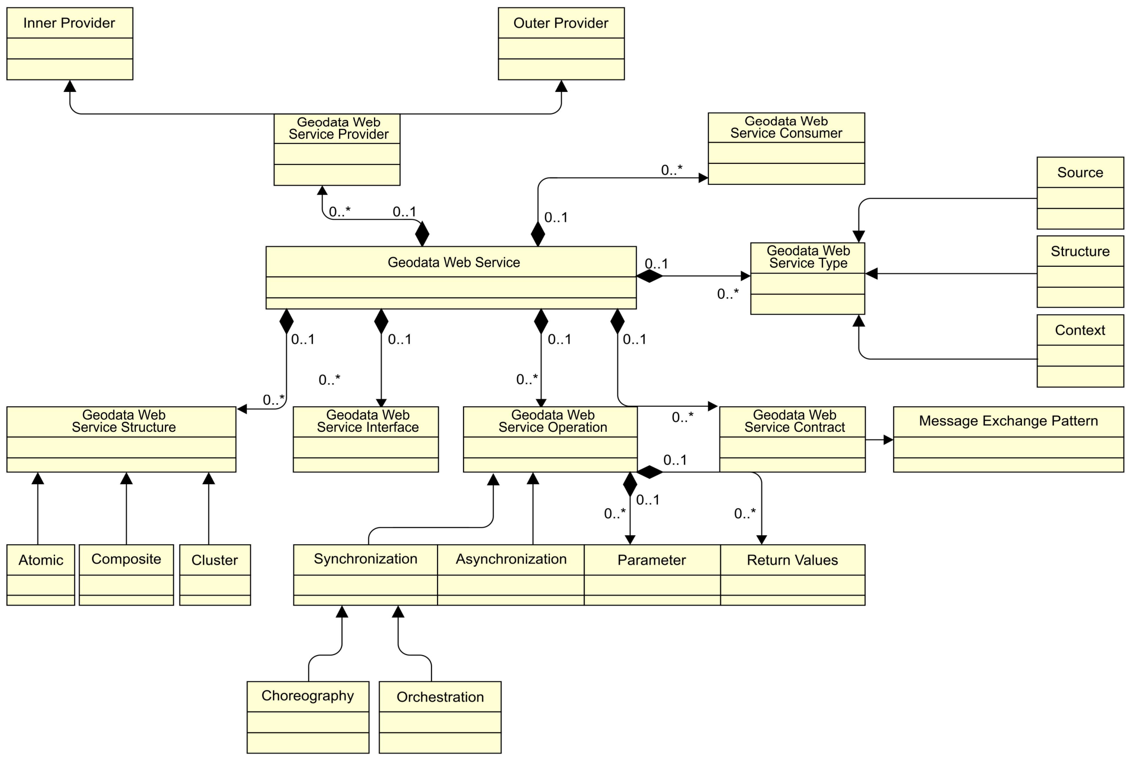

Figure 1. Basic geodata web service model.

Modelling of geodata web service composition requires descriptions of the services and service interactions that can be processed by computer. A geodata web service can be described based on a basic service description [36]. The properties of our geodata web service are described as follows:

- A geodata web service is an entity that has a life cycle. It is also a software solution that can solve business and technology problems and coexists with other software assets.
- A consumer is a geodata web service acquirer that participates in service interaction and is one of the participants. It cannot provide a service itself, but it acquires the service from peer services. The consumer interface is one of the service interfaces.
- A provider is a geodata web service provider that participates in service interaction and is also a participant. The provider interface is also one of the service interfaces.
- Geodata web service types can be classified into three categories: service source, service structure, and service context.

- Geodata web service structures can be classified into three basic service categories: atomic, composite, and cluster.
- Geodata web service interfaces define how the operations provided by the service are executed. They can be classified into simple and collaborating interfaces.
- A geodata web service contract provides many specifications to describe services and is more complex than a service interface. Services communicate with each other in accordance with the contracts, which contain all the information necessary for serving the providers and consumers [37].
- Messages are the information exchanged when geodata web services interact.

Participants are the entities, including people, systems, and applications, that participate in service interactions [37]. The functions provided by a service to consumers are contained in the service operation, which are implemented via request and response messages. Services can be collaborated into business processes. Figure 1 shows a basic geodata web service model based on the related work of service description.

2.2 Modelling geodata web service composition based on Petri nets

2.2.1 Methodology

Models based on Petri nets facilitate comprehension, definition, and analysis of composition behaviour in the preliminary and initial steps of their design. We use Petri nets as an underlying formalism in our work [19]. Petri nets and logic expressions are reviewed briefly below [38]. Service processes can be modelled as service nets.

Petri nets can be defined as follows [19]:

Definition 1: A Petri net (PN) is a tuple (P, T, F, M) in which

[i] P defines a set of places, $P = P_1 \cup P_2 \cdots \cup P_n$;

[ii] T defines a set of transitions, $T = T_1 \cup T_2 \cdots \cup T_n$;

[iii] F defines a set of arcs; and

[iv] M is the marking of PN , which describes the distribution of resources.

Definition 2: A service net (SN) is a tuple $(P, T, F, I, O, M, \mathrm{Pr})$ in which

[i] P defines a set of service places composed of service operation states and data being exchanged between the participants, $P = P_1 \cup P_2 \cdots \cup P_n$;

[ii] T defines a set of service transitions composed of service operations, $T = T_1 \cup T_2 \cdots \cup T_n$;

[iii] F defines a set of arcs of service processes, $SF \subseteq (SP \times ST) \cup (ST \times SP)$;

[iv] I defines a set of initial service process places;

[v] O defines a set of ending service process places;

[vi] $Pr: T \rightarrow (0,1] \times N^*$ is the firing probability and priority of transition, $Pr(t_i) = (\alpha_i, \beta_i)$, the default value of which is $(1,0)$; and

[vii] M is a marking function, and M_0 is the initial marking of PN.

Once the transition i under SN_i is fired, $M_i(p) = 1$. The element x in net SN_i is denoted by $SN_i \bullet x$. $\forall x \in (P \cup T)$, the pre-set of x, is denoted by $\bullet x = \{y \mid y \in (P \cap T) \wedge (y, x) \in F\}$, and the post-set of x is denoted by $x\bullet = \{y \mid y \in (P \cap T) \wedge (x, y) \in F\}$.

SN is mainly used to model services. Places describe the status and data control of services. Transitions represent the operation of services, and interfaces describe service input and output data. Mapping of interfaces is performed to realize interaction between services, with the terms α_i and β_i being used to describe the reliability of service. We can control the execution of a service by adjusting the priority of its transition under the same state.

2.2.2 Geodata web service composition structure

In general, procedural programming constructs have four basic patterns: sequential, parallel, loop, and conditional. These typical patterns, which are described below, can also be used to describe a geodata web service composition structure and can be used to form complex patterns. Geodata web service composition is accomplished based on service nets and the four basic patterns that address asynchronous and concurrent composite geodata service processes.

Sequential pattern: geodata web services are executed in sequence in a geodata web service composition based on Petri nets. Figure 2 illustrates a sequential pattern.

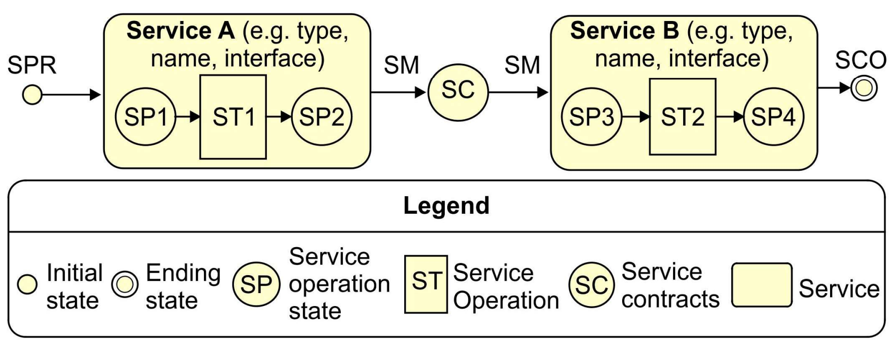

Figure 2. Sequential pattern for geodata web service composition.

Parallel pattern: geodata web services are executed in parallel in a geodata web service composition based on Petri nets. Figure 3 illustrates a parallel pattern. In this type of pattern, the geodata web service process is terminated only when all of the processes are completed.

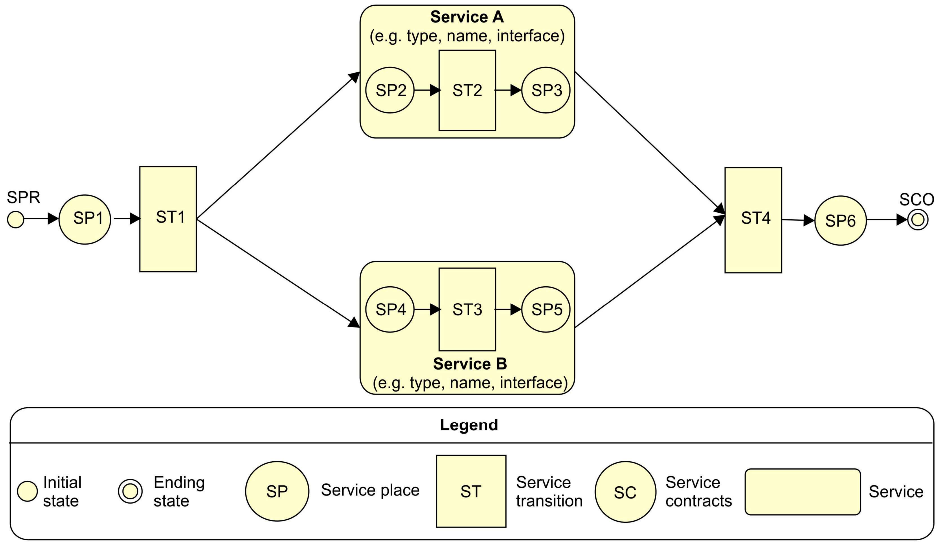

Figure 3. Parallel pattern for geodata web service composition.

Conditional pattern: geodata web services are executed conditionally in a geodata web service composition based on Petri nets. Figure 4 illustrates a conditional pattern. Once the service state place is satisfied, service A is executed; otherwise, alternative service B is executed.

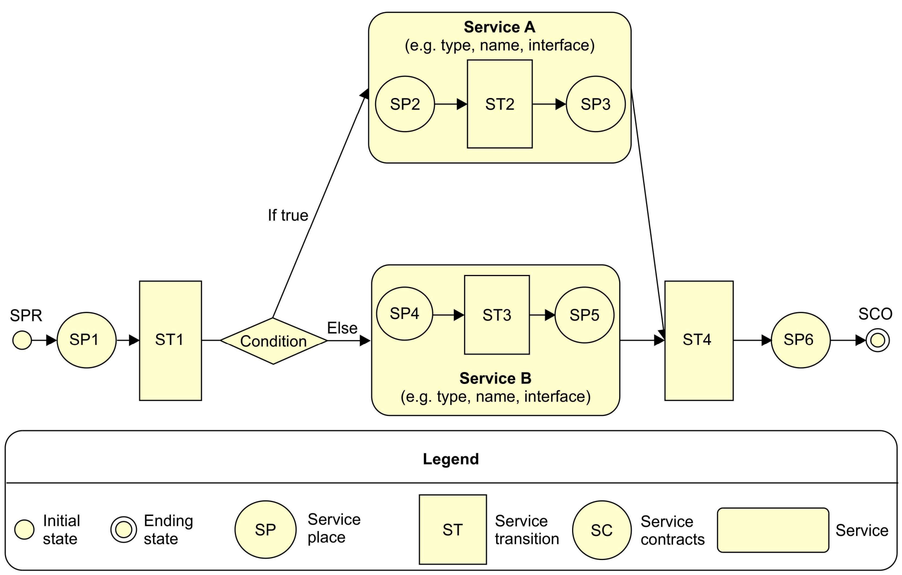

Figure 4. Conditional pattern for geodata web service composition.

Loop pattern: geodata web services are executed in a loop pattern in a geodata web service composition based on Petri nets. Figure 5 illustrates a loop pattern. Service A performs cyclically when the service place state is satisfied.

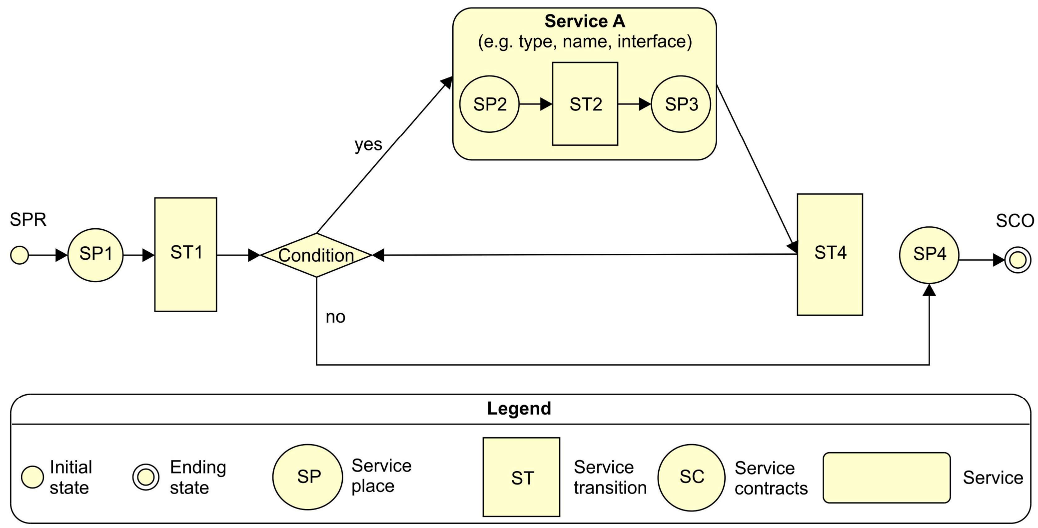

Figure 5. Loop pattern for geodata web service composition.

2.2.3 Example scenarios

The following scenarios exemplify geodata web service composition. One scenario is a combination of a geodata query service and a geodata delivery service. The other is a combination of a geomodel data acquisition service, a geomodel property acquisition service, and a geological model cutting service. The brief summaries of these scenarios provided below are sufficient to illustrate the proposed model and analysis approach.

As figure 6 shows, a user waits for geodata delivery before sending a request, while the geodata delivery service waits for a request before sending the geodata. It is assumed that the two parties have compatible interfaces for invoking geodata delivery and requesting geodata information and that the messages for the interfaces are also the same. The interaction between the two services may lead to a state of deadlock. How can we verify the deadlock? Process flow deadlock consists of six states: ‘start’, ‘request’, ‘response’, ‘unresponse’, ‘error’, and ‘end’. Each state represents one synchronous response call: ‘request’ represents a query for information, ‘response’ represents the process response that the called process sends back to the requestor, and ‘unresponse’ represents no response being sent, which will lead to endless waiting in the calling process and eventually result in deadlock. If the request sent by GeodataQuery does not receive a timely response from GeodataDelivery, a deadlock state will result. After geodata can be delivered, the cutting operation will be initiated. Figures 7 and 8 show the geodata web service composition being modelled based on Petri nets in accordance with the methodology described above. Figure 8 is a modified form of a figure we presented in a previous study

[36]. GeodataQuery, GeodataDelivery, geomodel data acquisition, geomodel property acquisition, and a geological model cutting can be modelled as a service net. A sequential pattern, a parallel pattern, and a loop pattern have been used to address them.

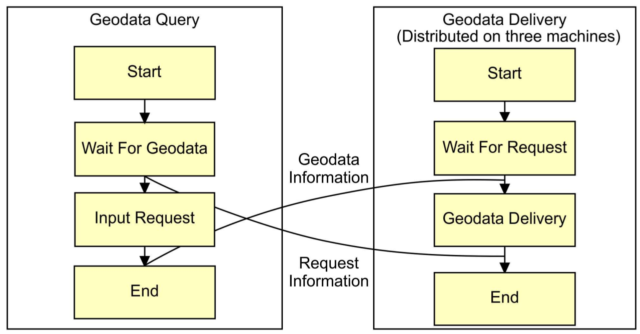

Figure 6. Geodata query and delivery business process.

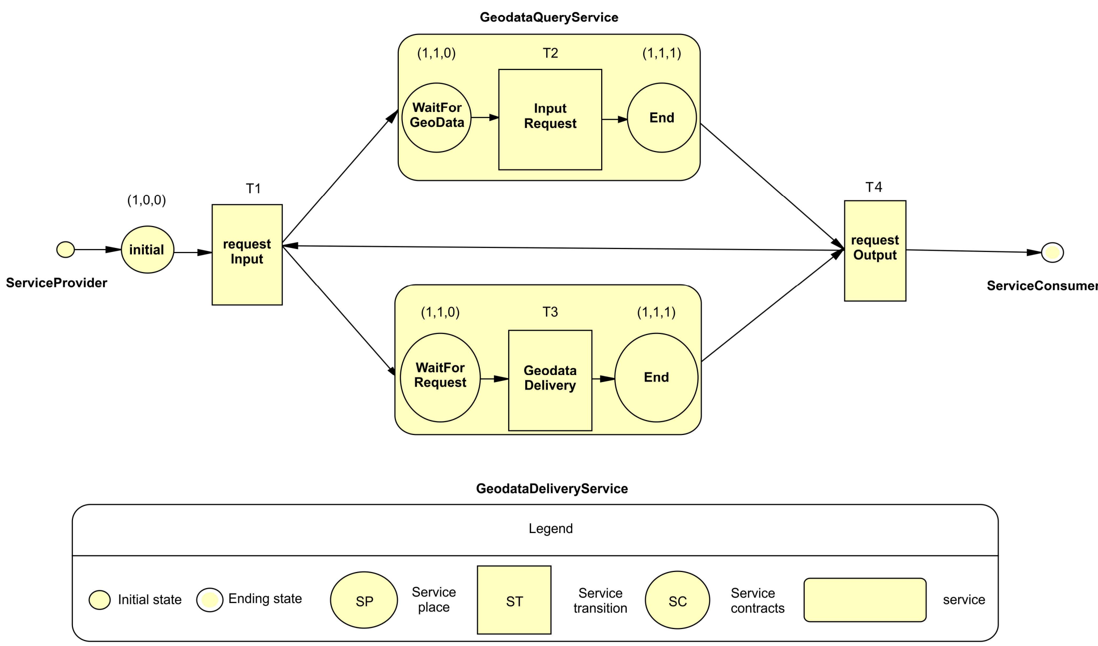

Figure 7. Modelling geodata query web service composition.

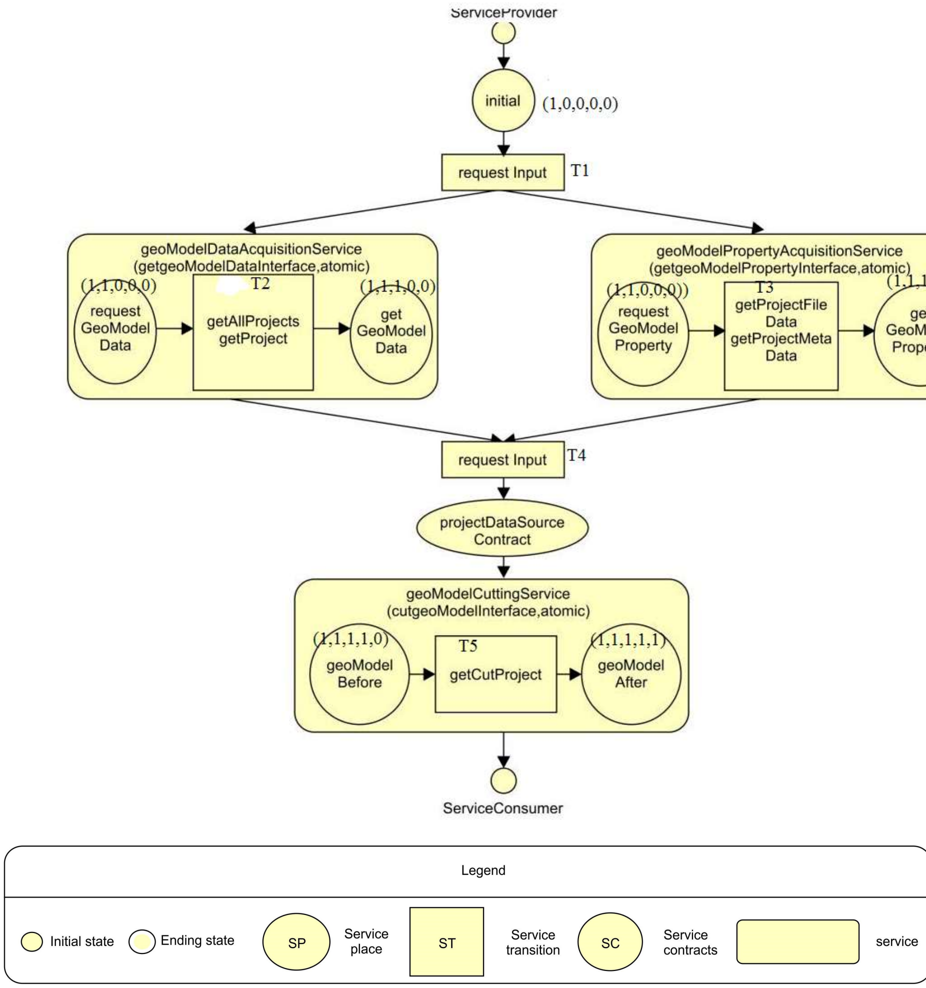

Figure 8. Modelling geodata cutting web service composition.

3 Analysis of geodata web service composition based on Petri nets

The Petri nets firing rule is associated with a graph that constitutes a representation of the nets' behaviour. The mapping interface can accomplish the interaction between services, i.e., the GeodataQuery service SN_1, the GeodataDelivery service SN_2, the GeomodeldataAcquisition service SN_3, the GeomodelpropertyAcquisition service SN_4, and the GeologicalmodelCutting service SN_5. Because the net is a bipartite graph, the verification is based on construction of the graph $G(SN_1, SN_2)$, $G(SN_3, SN_4, SN_5)$.

3.1 Reachability analysis

Reachability is defined as follows:

Definition 3: Reachability: $S = (M, TP)$ is called a state of a geodata web service composition model, where

[i] M describes the resource distribution, and

[ii] TP is the reachability probability of state S.

We say that the firing of transition t_i under state S is reachable.

Let S be a state of the geodata web service composition model. If the weight of $F(P_j, t_i)$ is W, for $t_i \in T$, transition t_i is enabled under state S if there are at least W tokens in each of its input places P_j. That is, given a transition t_i and the set of its input places $P_1, P_2, ...P_n$, transition t_i is enabled if $P_j \in \bullet t_i \rightarrow M(p_j) \geq W(p_j, t_i)$. All of the transitions enabled under state S are denoted by $ET(S) = \{t_i \mid S[t_i >, t_i \in T\}$. If there are many transitions under state S, choose the optimal transition t_i using Algorithm 1.

According to the definition of reachability, $S = (M_0, TP)$ once transition t_i can be fired, the resource token can be distributed in each P_i under S_i. As shown in figure 7, for GeodataQuery service SN_1, $M_1(P_2) = (1,1,0), M_1(P_3) = (1,1,1)$, and $ET(S_1) = t_2$; for GeodataDelivery service SN_2, $M_2(P_2) = (1,1,0), M_2(P_3) = (1,1,1)$ and $ET(S_2) = t_3$. As shown in figure 8, for GeomodeldataAcquisition service $SN3$, $M_1(P_2) = (1,1,0,0,0), M_1(P_3) = (1,1,1,0,0)$, and $ET(S_3) = t_2$; for GeomodelpropertyAcquisition service $SN4$, $M_2(P_4) = (1,1,0,0,0), M_2(P_5) = (1,1,1,0,0)$, and $ET(S_4) = t_3$; and for the GeologicalmodelCutting service $SN5$, $M_3(P_6) = (1,1,1,1,0), M_3(P_7) = (1,1,1,1,1)$, and $ET(S_5) = t_4$.

Following analysis of the state of the geodata web services composition model, from $M_1(P_3) = (1,1,1), ET(S_1) = t_2, M_2(P_3) = (1,1,1), ET(S_2) = t_3$, the reachability of $G(SN_1, SN_2)$ can be obtained. Similarly, from $M_3(P_6) = (1,1,1,1,0), M_3(P_7) = (1,1,1,1,1), ET(S_5) = t_4$, the reachability of $G(SN_3, SN_4, SN_5)$ can be obtained.

3.2 Boundedness analysis

Given a geodata web service composition, we can derive $G(SN_1, SN_2)$ and $G(SN_3, SN_4, SN_5)$, where SN_1 denotes the GeodataQuery service, SN_2 denotes the GeodataDelivery service, $SN3$ denotes the GeomodeldataAcquisition service, $SN4$ denotes the GeomodelpropertyAcquisition service, and $SN5$ denotes the GeologicalmodelCutting service, as shown in figures 7 and 8. $G(SN_1, SN_2)$, $G(SN_3, SN_4, SN_5)$ is a strongly connected graph; thus $G(SN_1, SN_2)$, $G(SN_3, SN_4, SN_5)$ are bounded.

Assuming that $G(SN_1, SN_2)$ is not a strongly connected graph, there exists an arc in at least one SN_i leading to another arc in another SN_i. This arc is an arc $t \rightarrow p$. Because every SN_i is live, there is a firing sequence of SN_i with infinite occurrences of t. By definition, place t's marking increases infinitely during this firing sequence, and thus $G(SN_1, SN_2)$ is unbounded.

Arc $p \to t$ is another arc. There is a firing sequence for $G(SN_1, SN_2)$ that includes infinite occurrences of t. If we project this sequence onto transitions of SN_i, this projected sequence is a firing sequence. By definition, all tokens p are distributed to transition t by a finite subsequence of the firing sequence. After n firings of t, the marking reached is m. Because t consumes at least one token of p, where $m(p) \geq m + n > n$, $G(SN_1, SN_2)$ is unbounded. Because $G(SN_1, SN_2)$ is a strongly connected graph, $G(SN_1, SN_2)$ is bounded. Similarly, $G(SN_3, SN_4, SN_5)$ is bounded.

3.3 Deadlock analysis

The definition of a service net path is as follows:

Definition 4 (service net path). C is a service net path leading from node n_i to node n_j if there exists a node sequence such that ($n_i \ldots\ldots n_j$). C denotes the path of ($n_i \ldots\ldots n_j$), where n_j is an inheritor of n_i in C.

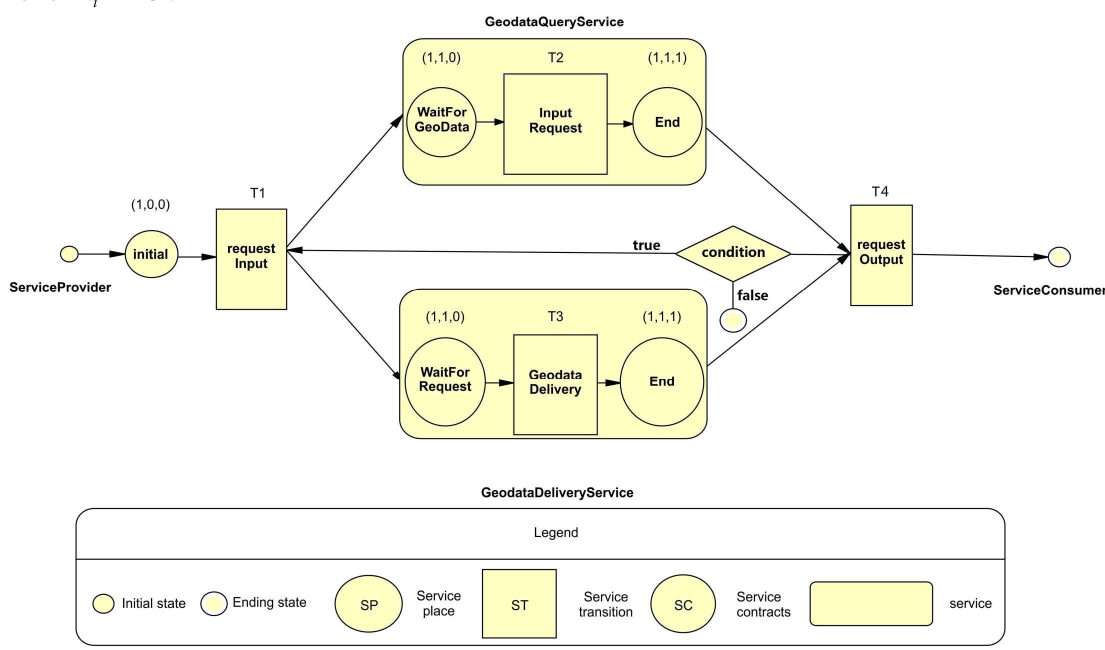

Figure 9. Modelling geodata query web service composition.

Given a graph $G(SN_1, SN_2)$ of a geodata web service composition, deadlock is caused by abnormal mergence of dual data places. As shown in figure 7, there are two patterns for the given service net SN_1, SN_2 and their dual data places $P_i(SN_i)$ and $P_j(SN_j)$. Only the path $\{C(P_i(S_i), P_j(S_j)), C(P_j(S_j), P_i(S_i))\}$ may lead to interactive deadlock. As figure 7 shows, for $\forall P_i, P_j \in S_i, S_j$, there exists $\{C(P_i(S_i), P_j(S_j)), C(P_j(S_j), P_i(S_i))\}$, and so there is deadlock in the geodata web service composition. If no $\{C(P_i(S_i), P_j(S_j)), C(P_j(S_j), P_i(S_i))\}$ existed, there would be no deadlock. Figure 8 shows no deadlock. The token P_i increases by one, and transition t_i generates a token, which is consumed immediately. Hence, the composition does not cause accumulation of tokens in any place. Thus, there is deadlock in the geodata web services composition. However, if we add the response condition and reply time, deadlock can be avoided. The Geodata web service composition modelling is shown in figure 9.

3.4 Optimization Analysis

Assume that GeodataDelivery is a service distributed on three machines, 1, 2, and 3, as shown in figure 6, based on the geodata characteristics. Which machine should be selected for optimality? In this section, a heuristic search algorithm is presented for geodata web service selection in geodata web service composition based on state analysis. Its search process is guided by a heuristic function based on the processing time of each individual service.

Algorithm. Algorithm A^* [39] is widely used for path finding and graph traversal. It uses a best-first search approach and finds the least-cost path from a given initial node to a goal node. In A^*, the heuristic cost function of current node x, denoted by $f(x)$, is used to guide the search process. The function $f(x)$ is the sum of two functions, $g(x)$ and $h(x)$, and $f(x) = g(x) + h(x)$, where $g(x)$ is the past path-cost function—which is the known cost from the initial node to the current node x and $h(x)$ is the future-path cost function, which is a heuristic estimate of the cost from x to the goal node. The procedure used by algorithm A^* based on the state is as follows.

[i] Put the initial marking m_0 on a list called 'first'.

[ii] Create another list called 'second' that is initially empty.

[iii] While (first $\neq \varnothing$),

(a) Select the first marking m_0 in 'first', move it to 'second'.

(b) If m_0 is the goal node, exit successfully.

(c) Seek all child nodes of m_0, and for every child node (denoted by x), calculate $g(x)$, $h(x)$, and $f(x)$.

(d) If x is not in 'first' or 'second', add x to 'first'.

(e) If x is already in 'first' or 'second', compare $f(x)$ with the new value of f on x and update $f(x)$ with the small value. If x is already in 'second', move x to 'first'.

[iv] Reorder 'first' in increasing magnitude of f.

Given $G(SN_1, SN_2)$, from the initial marking M_0, we can obtain many new markings $M_1 M_2$, as shown in figure 7, as the number of transitions enabled increases. From each new marking, we can again reach more markings. This process can result in a reachability graph $G(SN_1, SN_2)$. Under the same state S_2, the GeodataDelivery service is distributed on three machines, 1, 2, and 3. However, the question still remains as to which one is the optimal selection. This problem can be transformed into a search problem over the whole reachability graph. The schedule is the sequence of all transitions in a path from $M_1 M_2$ in the reachability graph. The reachability graph of this simple $G(SN_1, SN_2)$ is too large to generate in its entirety in this paper. Therefore, we assume here that the geodata delivery service distributed on machines 1, 2, and 3 is generated and searched. The function $g(x)$ is the actual time from the initial node to the current node x, and the heuristic function $h(x)$ is the time from node x to the goal node. It is well known that if M' is reachable from a marking M, there is a transition sequence such that $M[\alpha > M'$. In this paper, in accordance with Algorithm 1, the minimum processing time of a firing sequence of transitions from M_1 to M_2 is distributed on machines 1, 2, 3. We can conclude that the GeodataDelivery service is optimal.

4 Conclusions and Future Work

In this paper, we propose a procedure for the soundness verification of geodata web service composition based on Petri nets. The procedure uses Petri nets to validate the soundness of the geodata web service composition, and makes it possible to discover potential defects during the geodata web service composition design phase. The geodata web service composition produced using this approach can appropriately handle unexpected events that may occur at run time. This can shorten the defect discovery time, which reduces the redeployment cost, increases process reliability, and yields better overall results at a lower cost.

In this study, a geodata web service composition model was built based on Petri nets for dealing with asynchronous and concurrent composite service processes. A procedure for analysing the soundness of geodata web services composition in terms of reachability, boundedness, deadlock, and optimization analysis was also developed.

In future work, we plan to address issues associated with design patterns in complex service architectures. We also plan to undertake the challenging task of geodata web service composition model optimization using cloud computing [27].

Acknowledgements

We would like to thank the guest editors and anonymous reviewers for their comments and suggestions that have helped to improve this manuscript. This work was funded by the National Natural Science Foundation of China (Grant Nos. 61202363, 41202238, and U1261203).

References

1. Delipetrev, B., Jonoski, A. and Solomatine, D. P., Development of a web application for water resources based on open source software. Computers & Geosciences 62, 2014, 35-42.
2. Percivall, G., OGC Reference Model. Document Number OGC03-040 Version: 0.1.3. Technical Report, Open Geospatial Consortium. http://portal.opengeospatial.org/files/?artifact_id=3836S. Last accessed November 23, 2015.
3. Vretanos, P. (Ed.), Web Feature Service (1.0). OGC 02-058. Online: https://portal.opengeospatial.org/files/?artifact_id=7176. Last accessed November 23, 2015.
4. Evans, J. D. (Ed.), Web Coverage Service (WCS), version 1.0.0. OGC 03-065r6. Online: https://portal.opengeospatial.org/files/?artifact_id=3837. Last accessed November 23, 2015.
5. Kolodziej, K., OpenGIS Web Map Server Cookbook. Online: http://portal.opengeospatial.org/files/?artifact_id=7769. Last accessed November 23, 2015.
6. Schut, P., OpenGIS web processing service. Open Geospatial Consortium Inc., Wayland, MA, USA, 2007, 87.

7. Evangelidis, K., Ntouros, K., Makridis, S. and Papatheodorou, C., Geospatial services in the Cloud. Computers & Geosciences 63, 2014, 116-122.
8. Kiehle, C., Heier, C. and Greve, K., Requirements for next generation spatial data infrastructures—standardized web based geoprocessing and web service orchestration. Transactions in GIS 11, 2007, 819-834.
9. Cepicky, J. and Becchi, L., Geospatial processing via Internet on remote servers—PyWPS. OSGeo Journal, 2007, 1-5.
10. Brauner, J. and Schaeffer, B., Integration of GRASS functionality in web based SDI service chains. in Proceedings of FOSS4G 2008, Cape Town, South Africa, 2008, 420-429.
11. Fenoy, G., Bozon, N. and Raghavan, V., ZOO-Project: The open WPS platform. Applied Geomatics 5, 2013, 19-24.
12. WfMC, Process Definition Interface—XML Process Definition Language. Workflow Management Coalition, Cohasset, USA, 2008, p. 217.
13. Adams, M., ter Hofstede, A. H. M. and La Rosa, M., IMPACT open source software for workflow management: The case of YAWL, IEEE Software, vol. 28, 2011, 16-19.
14. Hertis, M. and Juric, M. B., An empirical analysis of business process execution language usage, IEEE Transactions on Software Engineering, vol. 40, 2014, 738-757.
15. Natschlaeger, C., Kossak. F. and Schewe, K., Deontic BPMN: A powerful extension of BPMN with a trusted model transformation, Software and Systems Modeling, vol. 14, 2015, 765-793.
16. Ghannoudi, M. and Chainbi, W., Formal verification for web service composition: A model-checking approach, 2015 International Symposium on Networks, Computers and Communications (ISNCC 2015), 2015.
17. Clempner, J. B., Structure vs. trajectory tracking methods: Soundness verification of business processes, IEEE Latin America Transactions, vol. 13, 2015, 3980-3986.
18. Bhardwaj, K. C. and Sharma, R. K., Machine learning in efficient and effective web service discovery, Journal of Web Engineering, vol. 14, 2015, 196-214.
19. Diaz, M., Petri Nets: Fundamental Models, Verification and Applications. John Wiley & Sons, Inc., 2013, 613.
20. Bramantoro, A., Ben Hassine, A., Matsubara, S. and Ishida, T., Multilevel analysis for agent-based service composition, Journal of Web Engineering, vol. 14, 2015, 63-79.
21. Cardinale, Y., El Haddad, J., Manouvrier, M. and Rukoz, M., CPN-TWS: A coloured Petri-net approach for transactional-QoS driven Web Service composition. IJWGS 7(1), 2011, 91-115.
22. Yu, H., Fan, G., Chen, L. and Liu, D., Analyzing time constrained service composition based on Petri net. in 3rd International Symposium on Electronic Commerce and Security Workshops, 2010, 68-71.
23. Fang, X., Jiang, C. and Fan, X., Independent global constraints for web service composition based on GA and APN. in Proceedings of the First ACM/SIGEVO Summit on Genetic and Evolutionary Computation, 2009, 119-126.
24. Lohmann, N., Massuthe, P., Stahl, C. and Weinberg, D., Analyzing interacting WS-BPEL processes using flexible model generation. Data Knowledge Engineering, 64(1), 2008, 38-54.
25. Du, Y., Li, X., and Xiong, P., A Petri net approach to mediation-aided composition of web services. IEEE Transactions on Automation Science and Engineering, vol. 9, 2012, 429-435.
26. Chi, Y.-L. and Lee, H.-M., A formal modeling platform for composing web services. Expert Systems Applications, 34(2), 2008, 1500-1507.
27. Xia, Y., Liu, Y., Liu, J. and Zhu, Q., Modeling and performance evaluation of BPEL processes: A stochastic Petri net-based approach. IEEE Transactions on Systems, Man, and Cybernetics, Part A 42(2), 2012, 503-510.
28. Blanco, E., Cardinale, Y. and Vidal, M.-E., Aggregating functional and non-functional properties to identify service compositions, IGI Book, 2011, 1-36.
29. Bonchi, F., Brogi, A., Corfini, S. and Gadducci, F., Compositional specification of web services via behavioural equivalence of nets: A case study. in van Hee, K.M. and Valk, R. (eds.) Petri Nets 2008. LNCS, Springer, vol. 5062, 2008, 52-71.
30. Hoisl, B., Sobernig, S. and Strembeck, M., Modeling and enforcing secure object flows in process-driven SOAs: An integrated model-driven approach, Software and Systems Modeling, vol. 13, 2014, 513-548.
31. Valero, V., Macía, H., Pardo, J.J., Cambronero, M.E., Díaz, G., Transforming web services choreographies with priorities and time constraints into prioritized-time colored Petri nets. Science of Computer Programming 77(3), 2012, 290-313.

32. Li, B., Xu, Y., Wu, J. and Zhu, J., A Petri-net and QoS based model for automatic web service composition. Journal of Software 7(1), 2012, 149-155.
33. Li, B., Ji, S., Qiu, D. and Cai, J., Generating test cases of composite services based on OWL-S and EH-CPNs, International Journal of Software Engineering and Knowledge Engineering, vol. 20, 2010, 921-941.
34. Cardinale, Y. and Rukoz, M., Fault tolerant execution of transactional composite web services: An approach. in Proceedings of UBICOMM, Lisbon, Portugal, 2011, 1-6.
35. Mei, X., Jiang, A., Li, S., Huang, C., Zheng, X. and Fan, Y., A compensation paired net-based refinement method for web services composition. Advances in Information Sciences and Service Sciences 3(4), 2011, 169-181.
36. Xu, N., Peng, S. and Wang, Z., Designing Geodata Service Composition Web Application Based on Service-Oriented Architecture, IEEE Access, vol. 4, 2016, 4136-4147.
37. Erl, T., Service-Oriented Architecture: Concepts, Technology, and Design. Prentice Hall, 2005, 792.
38. Hruz, B. and Zhou, M.C., Modeling and Control of Discrete Event Dynamic Systems, Springer, 2007.
39. Pearl, J., Heuristics: Intelligent Search Strategies for Computer Problem Solving. Addison-Wesley, 1984.