Sichere und Zuverlässige Prozessausführung in Serviceorientierten Architekturen

Dieter Schuller

Multimedia Communications Lab (KOM),
Technische Universität Darmstadt, Germany
Dieter.Schuller@KOM.tu-darmstadt.de

Zusammenfassung Prozesse in Serviceorientierten Architekturen lassen sich durch Komposition von Services realisieren. Die dabei verwendeten Services sind jedoch nicht notwendigerweise allesamt innerhalb der eigenen Unternehmung vorhanden, sondern über Organisationsgrenzen hinweg verteilt. Im Fall von fehlendem Einsatz entsprechender Sicherheitstechnologien sind die Nachrichten, die für die Invokation externer Services mit dem Service Provider ausgetauscht werden, möglicherweise das Ziel von Angreifern, die die Nachrichten manipulieren oder den Nachrichtenaustausch gänzlich verhindern können. Insofern sind Mechanismen erforderlich, die eine sichere und zuverlässige Serviceausführung ermöglichen. Der in dieser Arbeit vorgestellte Serviceüberwachungs- und Steuerungsansatz soll den Ausfall des gesamten Prozesses aufgrund von manipulierten, fehlerhaften oder ausgefallenen Services verhindern.

1 Einleitung

Geschäftsprozesse werden heutzutage nicht mehr ausschließlich innerhalb der Grenzen der eigenen Organisation ausgeführt (vgl. [1, 2]). Unternehmensübergreifende Prozesse gewinnen zunehmend an Bedeutung (vgl. [3]). Der Einsatz von Services – die je nach Granularität eine mehr oder weniger komplexe Funktionalität zur Verfügung stellen (vgl. [4]) – zur Realisation der Prozesse im Rahmen von Serviceorientierten Architekturen kann hier insofern nützlich sein, als er die Integration der verschiedenen Legacy Systeme und IT Systeme unterstützt. Externe Services, die von Geschäftspartnern entsprechend ihrer Kernkompetenzen angeboten werden oder auf Service-Marktplätzen vorhanden sind, können somit zur Effizienzsteigerung hinsichtlich der eigenen Prozessausführung eingebunden werden. Fällt einer der eingesetzten Services aus, kann aufgrund der Eigenschaft der losen Kopplung bspw. der fehlerhafte Service durch andere Services ersetzt werden, die die gleiche bzw. eine vergleichbare Funktionalität haben, sofern solche alternativen Services innerhalb der Unternehmung oder bei anderen Geschäftspartnern vorhanden bzw. auf Service-Marktplätzen verfügbar sind. Des Weiteren besteht jedoch die Gefahr, dass der für die Invokation externer Services notwendige Nachrichtenaustausch durch Angreifer manipuliert oder behindert wird (vgl. [5]). Um zu vermeiden, dass die Prozessausführung aufgrund der Invokation manipulierter Services erfolglos abbricht, sind Sicherheitsmechanismen erforderlich, die erkennen, ob ein Service manipuliert wurde.

Daher wird in Abschnitt 2 der in unserer Arbeit in [6] vorgestellte Ansatz zur Gewährleistung einer zuverlässigen Prozessausführung um Sicherheitsaspekte mit dem Ziel erweitert, eine sichere und zuverlässige Prozessausführung zu erreichen. Diesbezüglich wird einen Ansatz beschrieben, um *Sicherheit* für die Entscheidung zu quantifizieren, welcher Service für die Realisation eines bestimmten Prozessschrittes selektiert werden soll. Das entsprechende Optimierungsproblem wird in Abschnitt 3 formuliert und gelöst. Die Arbeit schließt mit einer Zusammenfassung und einem Ausblick.

2 Steuerungsansatz für sichere und zuverlässige Prozesse

Um einen Prozessausfall aufgrund von Servicefehlern zu verhindern, wurde ein dreistufiger Überwachungs- und Steuerungsansatz realisiert, der in Abbildung 1 dargestellt ist.

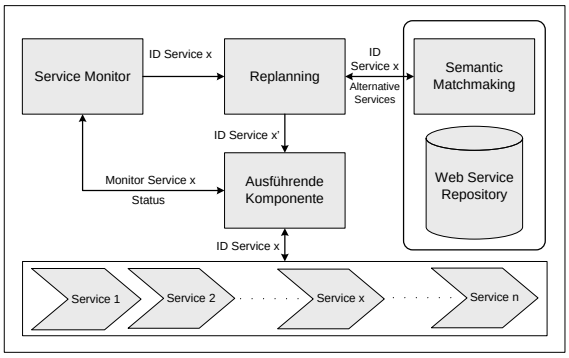


Abbildung 1: Replanning Zyklus

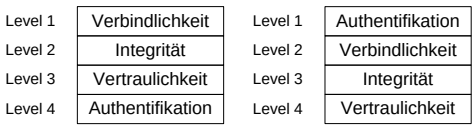
Der Service Monitor misst die Ausführungszeit des invokierten Services. Überschreitet dieser den in den Service Level Agreements (vgl. [7]) angegeben Wert, wird die Replanning-Komponente aufgerufen. Diese identifiziert unter Zuhilfenahme einer Semantic Matchmaking Komponente alternative Services (anhand von semantischen Annotationen), die im Web Service Repository gelistet sind. Um eine Auswahl zu treffen, welcher der alternativen Services den ausgefallenen ersetzen soll, stellt die Replanning-Komponente im Rahmen des Service-Selektions-Problems (vgl. [8–10]) das in Abschnitt 3 beschriebene Optimierungsproblem auf und löst es optimal hinsichtlich der nicht-funktionalen Service-Eigenschaften (Quality of Service – QoS). Dabei werden sowohl der bisher noch nicht erfolgreich ausgeführte Prozessschritt sowie alle nachfolgenden Prozessschritte des ursprünglichen Ausführungsplans für die Optimierung berücksichtigt.

Auf diese Weise ist jedoch noch nicht sichergestellt, dass die eingebundenen Services nicht von Angreifern manipuliert wurden. Insofern sind Sicherheitsmechanismen wie Verschlüsselung, Digitale Signaturen und Checksummen erforderlich,

die eine solche Manipulation erkennen, um den manipulierten Service durch einen alternativen Service (wie beschrieben) auszutauschen. Um im Rahmen des Service-Selektions-Problems diese qualitativen Sicherheitseigenschaften von Services als QoS-Parameter berücksichtigen zu können, ist eine Quantifizierung dieser Eigenschaften notwendig. Eine Voraussetzung stellt dabei ihre Erfassbarkeit dar. Andernfalls lässt sich keine nachvollziehbare, konsistente Zuordnung der qualitativen Eigenschaft auf einen numerischen Wert finden. Konsistent bedeutet dabei, dass einer erfassten Eigenschaft immer der gleiche, numerische Wert zugewiesen wird. Im Kontext von Sicherheitsmetriken definiert Jaquith in seiner Arbeit [11] eine gute Metrik als einen „konsistenten Standard für die Messung". Er fordert von einer guten Metrik, dass ihre Messung konsistent möglich sein sollte und dass sie günstig (vorzugsweise automatisiert) erfassbar ist. Zudem sollte sie durch eine numerische Zahl ausgedrückt werden können und nicht durch qualitative Kennzeichnungen wie „high", „medium" oder „low".

Die vorliegende Arbeit verfolgt das Ziel, einen nachvollziehbaren und konsistenten Ansatz zur Zuordnung von Sicherheitseigenschaften auf numerische Werte zu finden. Hierfür wird in Anlehnung an die in [12] definierten Schutzziele (Daten-) Integrität, (Informations-) Vertraulichkeit, Verfügbarkeit, Verbindlichkeit, Authentizität und Privatheit entsprechende *Sicherheits-Level* als QoS-Parameter definiert. Sind die mit externen Serviceanbiertern ausgetauschten Nachrichten bspw. verschlüsselt und/oder digital signiert, sind die Schutzziele Authentizität und Vertraulichkeit erfüllt. Zudem kann mithilfe von Checksummen die Integrität der erhaltenen Nachrichten überprüft werden. Je nachdem, welche Schutzziele von einem Service bzw. von einem Service Provider erfüllt werden, wird für den betreffenden Service ein gewisser Level gesetzt. D. h., dass bestimmte Sicherheits-Eigenschaften bestimmten Levels zugeordnet werden. Dies kann jedoch nicht generisch vorgenommen werden, sondern muss spezifisch für eine bestimmte Situation bzw. für ein bestimmtes Szenario durchgeführt werden. Bspw. könnte es in einem Business-Kontext wichtiger sein, dass zur Wahrung von Betriebsgeheimnissen die Informationen verschlüsselt sind, die zur Invokation eines externen Services als Eingabeparameter die Unternehmensgrenzen verlassen, als dass die zurück gelieferten Ergebnisse integer sind. Handelt es sich jedoch um erfolgskritische Prozesse (wie bspw. in einem Katastrophenszenario), sind Integrität und Autorisierung von essentieller Bedeutung. Allerdings ist die alleinige Zuordnung einer Sicherheits-Eigenschaft zu einem Sicherheitslevel nicht immer ausreichend. Bspw. könnte ein Angreifer das Ergebnis einer externen Service-Invokation abfangen und stattdessen andere Ergebnisse an den ursprünglichen Service-Aufrufer weiterleiten. Diese Ergebnisse sind zwar auch integer, sie stammen aber nicht von der Instanz, von der die Funktionalität des aufgerufenen Services erwartet wurde. Um hier ein Beispiel zu nennen, könnte ein Angreifer in einem Katastrophenszenario mit Hochwasser falsche Informationen über Pegelstände weiterleiten, sodass notwendige Maßnahmen nicht angemessen geplant werden können. In diesem Kontext muss also eine Kombination von Sicherheits-Eigenschaften vorliegen, um eine Zuordnung zu bestimmten Sicherheits-Levels vornehmen zu können. Als mögliche Kombinationen werden im Rahmen dieser Arbeit zunächst die in Abbil-

dung 2a und Abbildung 2b angegebenen Level vorgeschlagen. Die Level haben per Definition h ganzzahlige numerische Werte $s \in S = \{1, ..., h\}$. Dabei wird dieser Parameter als positiver QoS-Parameter definiert, sodass ein höherer Wert besser ist. Sind auf diese Weise Sicherheits-Level für die verwendeten Services erstellt worden, lässt sich dieser nun quantifizierte Parameter als QoS-Parameter für die Formulierung des Optimierungsproblems in Abschnitt 3 verwenden.



(a) Kombination 1 (b) Kombination 2

Abbildung 2: Sicherheits-Level

3 Optimierungsproblem

Die in Abschnitt 2 angesprochene Replanning-Komponente befasst sich mit dem Service-Selektions-Problem. D. h., sie erstellt einen optimalen Ausführungsplan, indem sie entscheidet, welcher Prozessschritt von welchem dafür infrage kommenden Service realisiert werden soll. Optimal bedeutet in diesem Zusammenhang, dass diejenigen Services selektiert werden, die vorgegebene Restriktionen einhalten und hinsichtlich ihrer QoS Eigenschaften am besten sind. In anderen Worten ist bei der Auswahl der Services zu beachten, dass die gebildete Servicekomposition einerseits bestimmte Restriktionen in Bezug auf ihre QoS-Parameter einhält, die in Modell 1 als Nebendingungen formuliert sind. Andererseits soll sie die in (2) angegebene Zielfunktion maximieren, um eine optimale Lösung darzustellen. Ein Beispiel für eine einzuhaltende Restriktion hinsichtlich der aggregierten Ausführungszeit der ausgewählten Services wäre eine obere Grenze von 20 Sekunden. Hinsichtlich der Zuverlässigkeit könnte eine untere Grenze bspw. 95%. Dabei hängen die unteren und oberen Schranken sowie die Zielfunktion von dem betrachteten Szenario ab.

Um das Systemmodell zu erstellen und das Optimierungsproblem zu formulieren, wird unser in [6] vorgestellte, generische Ansatz für die fünf QoS-Parameter Zuverlässigkeit r, Sicherheit s, Verfügbarkeit a, Ausführungszeit t und Kosten c, die in dieser Reihenfolge mit $k \in K = \{1, 2, 3, 4, 5\}$ nummeriert werden, erweitert. Dabei wird zunächst von einer sequenziellen Anordnung von n Prozessschritten ausgegangen. Prozessschritt $i \in I = \{1, ..., n\}$ wird vor Prozessschritt $i + 1$ ausgeführt. Für jeden Prozessschritt i gibt es m_i alternative Services $j_i \in J_i = \{1, ..., m_i\}$, wobei Prozessschritt i durch genau einen Service j_i realisiert wird. Diese Services j_i unterscheiden sich hinsichtlich ihrer fünf QoS-Parameter q_{ijk}. Ist ein höherer (geringerer) QoS-Wert besser, handelt es

sich um einen positiven (negativen) QoS-Parameter. Die Restriktionen für die QoS werden mit b_k bezeichnet. Die Entscheidungsvariablen $x_{ij} \in \{0,1\}$ geben Auskunft darüber, ob Prozessschritt i durch Service j realisiert wird. Um die (unterschiedlichen) QoS-Parameter für die Zielfunktion aggregieren zu können, ist eine Normalisierung für alle Werte q_{ijk} erforderlich, die in Gleichung (1) angegeben ist. Andernfalls lässt sich bspw. die Sicherheit als QoS-Parameter nicht mit der Zuverlässigkeit eines Services verrechnen. Das zu lösende Optimierungsproblem wird in Modell 1 formuliert. Vor dem Hintergrund, eine sichere und zuverlässige Prozessausführung zu erreichen, werden die QoS-Parameter Zuverlässigkeit r, Sicherheit s und Verfügbarkeit a mit jeweils $33,3\%$ für die Zielfunktion gewichtet.

$$q_{ijk}^{norm} := \begin{cases} 1 - \frac{max\{q_{ijk}\} - q_{ijk}}{max\{q_{ijk}\} - min\{q_{ijk}\}} & \text{, falls k negativer QoS} \\ \frac{q_{ijk} - min\{q_{ijk}\}}{max\{q_{ijk}\} - min\{q_{ijk}\}} & \text{, sonst} \end{cases} \tag{1}$$

Modell 1 Nicht-lineares Optimierungsproblem

Zielfunktion

$$\text{maximiere } F(x) = \sum_{i=1}^{n} \sum_{j=1}^{m_i} \sum_{k=1}^{3} \frac{1}{3} q_{ijk}^{norm} x_{ij} \tag{2}$$

s.d.

$$\prod_{i=1}^{n} \sum_{j=1}^{m_i} r_{ij} x_{ij} \geq b_1 \tag{3}$$

$$\min \left\{ \sum_{j=1}^{m_i} s_{ij} x_{ij} \right\} \geq b_2 \tag{4}$$

$$\prod_{i=1}^{n} \sum_{j=1}^{m_i} a_{ij} x_{ij} \geq b_3 \tag{5}$$

$$\sum_{i=1}^{n} \sum_{j=1}^{m_i} t_{ij} x_{ij} \leq b_4 \tag{6}$$

$$\sum_{i=1}^{n} \sum_{j=1}^{m_i} c_{ij} x_{ij} \leq b_5 \tag{7}$$

$$\sum_{j=1}^{m_i} x_{ij} = 1 \qquad \forall i \in I \tag{8}$$

$$x_{ij} \in \{0,1\} \quad \forall i \in I, \forall j \in J_i \tag{9}$$

Die zu maximierende Zielfunktion ist in (2) angegeben. In (3), (4), (5), (6) und (7) sind die Nebenbedingungen für die QoS-Parameter Zuverlässigkeit r,

Sicherheit s, Verfügbarkeit a, Ausführungszeit t und Kosten c angegeben. In (8) wird gefordert, dass für jeden Prozessschritt i genau ein Service ausgewählt wird. Die Ganzzahligkeitsbedingung in (9) stellt sicher, dass lediglich *ganze* Services selektiert werden.

Da es sich bei (3), (4), (5) um nicht-lineare Gleichungen handelt, wird für (3) und (5) die Approximation in (10) angewendet, die für z nahe 1 gute Werte liefert [13]. Des Weiteren lässt sich (4) durch (11) ersetzen.

$$\prod_{i=1}^{n}\sum_{j=1}^{m_i} z_{ij}x_{ij} \approx 1 - \sum_{i=1}^{n}(1 - \sum_{j=1}^{m_i} z_{ij}x_{ij}) \tag{10}$$

$$\sum_{j=1}^{m_i} s_{ij}x_{ij} \geq b_2 \qquad \forall i \in I \tag{11}$$

Das hierdurch erhaltene lineare Optimierungsproblem lässt sich mit Methoden des Operations Research optimal lösen, sofern eine Lösung existiert (vgl. [14]). Hierbei ist jedoch zu berücksichtigen, dass es sich bei dem beschriebenen Optimierungsproblem um ein NP-schweres Problem handelt [9, 15, 16]. Bei steigender Problemgröße lässt sich die Heuristik H1_RELAX_IP [17] verwenden, die bei großen Problemgrößen nicht signifikant schlechter abschneidet als das exakte Lösungsverfahren [18]. Hier wird die Ganzzahligkeitsbedingung relaxiert und das so entstandene gemischt-ganzzahlige lineare Optimierungsproblem optimal gelöst. Anschließend werden diejenigen Services ausgewählt, deren Werte in den Entscheidungsvariablen x_{ij} am größten sind.

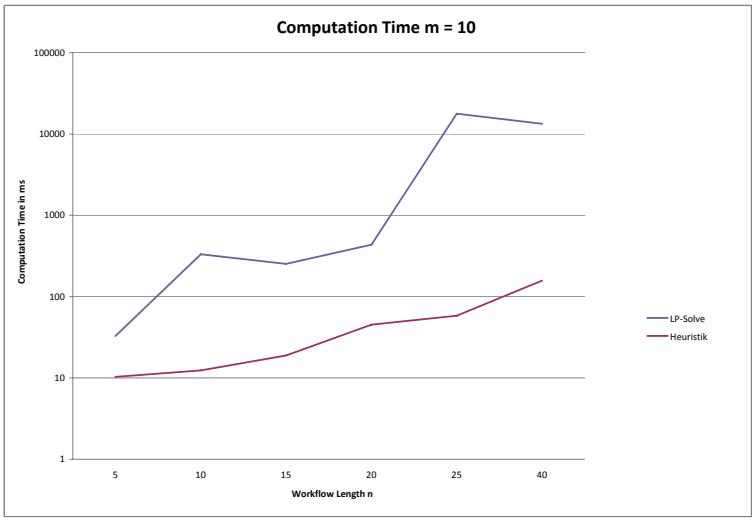


Abbildung 3: Berechnungszeit bei Variation der Prozesslänge

Um einen Einblick in die Laufzeitkomplexität zu geben, wird die Laufzeit für die Berechnung der optimalen Lösung (durch Einsatz von Branch-and-Bound

Verfahren) für das formulierte Optimierungsproblem in Abbildung 3 und Abbildung 4 mit der Laufzeit der Heuristik H1_RELAX_IP verglichen. Dabei erreicht die Heuristik zumeist eine Lösungsgüte von mehr als 95% der optimalen Lösung. In Abbildung 3 wird die Anzahl alternativer Services pro Prozessschritt m_i auf $m_i = 10$ fixiert und die Anzahl an Prozessschritten n variiert. In Abbildung 4 wird die Anzahl an Prozessschritten n auf $n = 10$ fixiert und die Anzahl alternativer Services pro Prozessschritt m_i variiert.

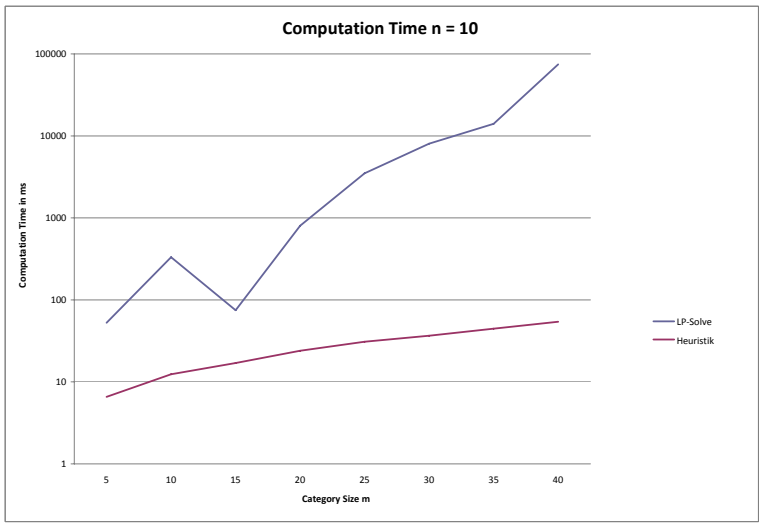


Abbildung 4: Berechnungszeit bei Variation der Anzahl alternativer Services pro Prozessschritt

4 Zusammenfassung und Ausblick

In der vorliegenden Arbeit wurde ein Serviceüberwachungs- und Steuerungsansatz vorgestellt, der durch eine optimale Lösung des Service-Selektions-Problems aufgrund der Gewichtung der QoS-Parameter Sicherheit, Zuverlässigkeit und Verfügbarkeit eine sichere und zuverlässige Prozessausführung gewährleisten soll.

Der Fokus für die zukünftige Arbeit wird einerseits auf die Formulierung von linearen Optimierungsproblemen für komplexe Prozesse gelegt, um den vorgestellten Ansatz auch für komplexe Prozessstrukturen verwenden zu können. Andererseits sollen Kombinationen der betrachteten Sicherheits-Eigenschaften hinsichtlich des erreichten Sicherheitsniveaus (Quality of Protection) analysiert werden, um ein Konzept für die Definition von sinnvollen Sicherheits-Levels zu entwickeln. Zudem wird die Evaluation des vorgestellten Ansatzes angestrebt.

Acknowledgements. Diese Arbeit wurde in Teilen durch das BMBF-finanzierte Projekt SoKNOS (http://www.soknos.de) und durch das E-Finance Lab e. V., Frankfurt am Main, Deutschland, (http://www.efinancelab.de) unterstützt.

Literatur

1. Bussler, C.: The Role of B2B Protocols in Inter-Enterprise Process Execution. In: International Workshop on Technologies for E-Services (TES). (2001) 16–29
2. Chen, Q., Chen, Q., Hsu, M., Hsu, M.: Inter-Enterprise Collaborative Business Process Management. In: International Conference on Data Engineering (ICDE). (2001) 253–260
3. Leymann, F., Roller, D.: Production Workflow: Concepts and Techniques. Prentice Hall PTR, Upper Saddle River, NJ, USA (2000)
4. Krafzig, D., Banke, K., Slama, D.: Enterprise SOA: Service-Oriented Architecture Best Practices. Prentice Hall PTR, Upper Saddle River, NJ, USA (2004)
5. Miede, A., Nedyalkov, N., Schuller, D., Repp, N., Steinmetz, R.: Cross-organizational Security – The Service-oriented Difference. In: International Conference on Service-Oriented Computing (ICSOC) – 2009 Workshops. (2009)
6. Schuller, D., Papageorgiou, A., Schulte, S., Eckert, J., Repp, N., Steinmetz, R.: Process Reliability in Service-oriented Architectures. In: Digital Ecosystems and Technologies (DEST). (2009) 606–611
7. Keller, A., Ludwig, H.: The WSLA Framework: Specifying and Monitoring Service Level Agreements for Web Services. Journal of Network and Systems Management **11**(1) (2003) 57–81
8. Jaeger, M.C., Mühl, G., Golze, S.: QoS-Aware Composition of Web Services: A Look at Selection Algorithms. In: Internationl Conference on Web Services (ICWS). (2005) 807–808
9. Yu, T., Lin, K.J.: Service Selection Algorithms for Composing Complex Services with Multiple QoS Constraints. In: International Conference on Service-oriented Computing (ICSOC). (2005) 130–143
10. Ardagna, D., Giunta, G., Ingraffia, N., Mirandola, R., Pernici, B.: QoS-Driven Web Services Selection in Autonomic Grid Environments. In: OTM Conferences (2). (2006) 1273–1289
11. Jaquith, A.: Security Metrics: Replacing Fear, Uncertainty, and Doubt. Addison-Wesley Professional (2007)
12. Eckert, C.: IT-Sicherheit. 5. edn. Oldenbourg Wissensch.Vlg (November 2007)
13. Heckmann, O.: A System-oriented Approach to Efficiency and Quality of Service for Internet Service Providers. PhD thesis, TU Darmstadt, Fachbereich Informatik (2004)
14. Domschke, W., Drexl, A.: Einführung in Operations Research. Springer Verlag, Heidelberg (2007)
15. Canfora, G., Penta, M.D., Esposito, R., Perfetto, F., Villani, M.L.: Service Composition (Re)Binding driven by Application-Specific QoS. In: 4th International Conference Service-Oriented Computing (ICSOC). (2006) 141–152
16. Zeng, L., Benatallah, B., Ngu, A.H., Dumas, M., Kalagnanam, J., Chang, H.: QoS-Aware Middleware for Web Services Composition. Transactions on Software Engineering **30**(5) (2004) 311–327
17. Berbner, R., Spahn, M., Repp, N., Heckmann, O., Steinmetz, R.: Heuristics for QoS-aware Web Service Composition. In: International Conference on Web Services (ICWS). (2006)
18. Berbner, R., Spahn, M., Repp, N., Heckmann, O., Steinmetz, R.: Dynamic Replanning of Web Service Workflows. In: Digital Ecosystems and Technologies (DEST). (2007)