

Research Challenges on Online Service Quality Prediction for Proactive Adaptation

Andreas Metzger*, Chi-Hung Chi†, Yagil Engel‡, Annapaola Marconi§

*Paluno, University of Duisburg-Essen, Essen, Germany, andreas.metzger@paluno.uni-due.de

†School of Software, Tsinghua University, Beijing, China, chichihung@mail.tsinghua.edu.cn

‡IBM Research Labs, Haifa, Israel, yagile@il.ibm.com

§SOA, FBK, Trento, Italy, marconi@fbk.eu

Abstract—Online quality prediction allows service-oriented systems to anticipate the need for adaptation and thus to prevent the actual occurrence of failures or to mitigate upcoming failures. Such proactive adaptation capabilities are increasingly relevant for future service-oriented systems, which need to cope with limited control over third-party services, as well as rapidly changing usage contexts. Initial, promising results have been achieved for what concerns online quality prediction for service-oriented systems. However, there are many challenging issues remaining that call for concrete solutions. In this paper we present a set of research challenges identified by the research community that may be worth investigating in the coming years.

Keywords-Failure prediction, quality prediction, SOA

I. INTRODUCTION

Service-orientation is increasingly adopted as a paradigm for building highly dynamic, distributed service-oriented systems. A service-oriented system is realized by composing individual software services. In contrast to a software component, not only the development, quality assurance, and maintenance of the software can be under the control of third-parties, but the software can also be executed and managed by third-parties [1].

There is a clear trend that in the future, service-oriented applications will be increasingly composed of *third-party services* accessible over the Internet [2]. As a consequence, the capabilities and quality of service-oriented systems more and more will depend on the quality of its third-party services. Specifically, this means that service-oriented systems have to become resilient against failures of their third-party services.

Furnishing service-oriented systems with self-adaptation capabilities is considered a key solution to address the above challenge [1], [3], [4]. *Online quality prediction*, together with self-adaptation, allows service-oriented systems to anticipate the need for adaptation and thus to prevent the actual occurrence of failures or to mitigate upcoming failures. Such a *proactive* adaptation of service-oriented systems promises to mitigate some of the key problems faced when resorting to reactive adaptation only, such as the need for costly repair or compensation activities (cf. [5]–[7]).

A. Need for Novel Online Quality Prediction Techniques

Dedicated, novel online quality prediction techniques are increasingly relevant for future service-oriented systems, due to the following two reasons:

1) *Limited Control*: The use of third-party services implies that service compositions will be subject to changes at runtime in their execution environment that are only under limited control by the service integrator [8], [9]. Examples include service changes, such as new versions of services that are incompatible with previous versions, the discontinuation of service offerings by their providers, as well as fluctuations in service quality, such as performance, availability and reliability. In most cases those changes and potential failures cannot be predicted with existing design time techniques and methods.

2) *Limited Visibility*: Due to the fact that a significant part of the software and the infrastructure that realize service compositions will be owned, hosted and maintained by third parties, service integrators will only have limited visibility into the intrinsics of the building blocks of those compositions. In most cases services can only be observed through the “behavioral” interface that is offered by the service provider. The architecture, control flow or even code of third-party services will not be known by service integrators. Service providers in most cases will not be willing to share this information [8]. This poses a critical limitation to the applicability of existing online failure prediction techniques, which have been proposed for more traditional computer-based systems (see [10] for a comprehensive survey).

B. State of the Art

The above reasons led researchers to propose online quality prediction techniques tailored to the realm of service-oriented systems. Those techniques fall into the following major classes:

- *Data mining* approaches, as presented in [11], [12], leverage machine learning capabilities to train prediction models using historic monitoring data. As an example, they employ multi-layer artificial neural networks [13] for quantitative QoS and decision trees [14] for qualitative QoS.
- *Run-time verification* approaches, as introduced in [15], [16], ascertain whether some predefined properties are

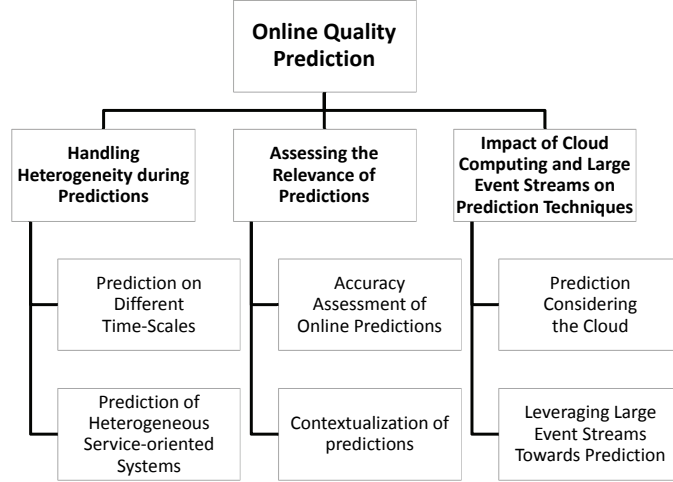


Figure 1. Overview on Research Challenges

met at run-time. A typical technique which is used is run-time model checking.

- *Online testing* approaches, as advocated in [7], [17], [18], test the services of the service-oriented system in parallel to their normal use in order to gather additional evidence for failures.
- *Static analysis* approaches, such as the ones introduced in [19], systematically examine models of service-oriented systems to infer approximations of their execution. They use the service composition structure of the system as a basis to forecast QoS deviations by mapping it to a constraint satisfaction problem (CSP) [20]
- *Simulation-based* approaches, as presented in [21], [22], execute dynamic models of the service-oriented systems to simulate their future behavior. Those approaches mostly resort to discrete event simulation tools [23].

Despite such initial progress in the field of online quality prediction for service-oriented systems, many issues still remain open and new issues will arise for future service-oriented systems.

C. Contribution of the Paper

This paper presents and elaborates on important research challenges for online quality prediction of service-oriented systems. Those challenges have been identified by 40 experts during the S-Cube Research Roadmap Workshop [24]. The overall objective of the workshop was to identify challenges that may become relevant after and beyond the S-Cube project (i.e., in 5–10 years) and which have a high potential for impact.

II. CHALLENGES FOR ONLINE QUALITY PREDICTION

The challenges on online quality prediction are grouped in to three main clusters and detailed below (also see Fig. 1):

- handling heterogeneity during predictions (Section II-A),
- assessing the relevance of predictions (Section II-B), and
- impact of cloud computing and large event streams on prediction techniques (Section II-C).

A. Handling Heterogeneity during Predictions

1) *Prediction on Different Time-Scales*: Adaptability is recognized as a key feature of service-oriented systems, where achieving optimal and/or resilient operation requires to continuously re-structure the system to adapt to changes in its execution context. These adaptation needs may be triggered by specific cases to be handled, unexpected situations depending on environmental conditions, or changing requirements.

This need for continuous adaptation results in a system characterized by a huge set of executions that, although instantiated from the same models (e.g., service composition specifications such as BPEL or BPMN), strongly differ in terms of run-time structure. In such a dynamic environment, the system cannot remain unchanged; the *short-term* adaptations applied to process and service instances should be used to derive *long-term* system changes, i.e., to predict beneficial evolutions of the system. In fact, providing support for long-term adaptation is becoming one of the main requirements for managing the life-cycle of dynamic systems [25]. In particular, success cases and best practices derived from the execution and adaptation history should be used as training cases for evolution mechanisms in order to progressively improve the system.

Most existing approaches addressing this problem (e.g., [26], [27]) derive model-level changes by analyzing frequently occurring changes at the execution-level. In other words, if a run-time change/adaptation occurs more frequently than a predefined threshold, the change will be propagated at the model-level. These evolution approaches

present two major drawbacks. First, an execution-level adaptation is not good “in general”, it is good for a specific context/situation, and thus cannot simply be propagated to the system model without taking into account the adaptation need it was devised for. Moreover, plugging in adaptation variants into the original system is not always a good solution, since it may result in embedding fault-handling activities rather than trying to address the root cause that required run-time adaptation. Few approaches have been proposed in this direction (e.g., [28], [29]), and the results are still preliminary.

Another key challenge not addressed so far is learning from the execution and adaptation history to automatically improve the learning rules/strategies that are used to proactively trigger adaptation and search for run-time adaptation solutions.

2) *Prediction of Heterogeneous Service-oriented Systems:* Made possible through advancements in information and communication technology, we will see the seamless integration of virtual services (such as financial and telecommunication services) with real-world services (such as transportation or manufacturing). Service-orientation will thus foster and ease cross-organizational data exchange and integration of IT systems. This trend will culminate in the Future Internet, where the Internet of Services (IoS) and the Internet of Things (IoT) converge.

The Future Internet thus will radically empower stakeholders and IT systems to access heterogeneous and cross-organizational services and data. Especially the IoT will lead to an unprecedented access to more data sources (e.g., through the availability of cheap connected sensors) and thus will foster the seamless and effortless access to operational data from everywhere at any time.

Such data availability and online access to the data, as well as novel means to interact with the virtual and real-world services, will open up opportunities for innovative ways of monitoring and predicting and therewith controlling, adapting and managing business processes and business interactions.

For the vision of the Future Internet to become true, many challenging research issues need to be addressed. Important ones include:

- We will see an even stronger decentralization of systems together with the lack of control of such decentralized entities. This requires novel ways to observe these systems in order to gain evidence for prediction.
- Due to many different stakeholders and data sources, there is an increased need to understand the quality of that data (cf. [30]). Here, issues such as accuracy and timeliness of data from the IoT, as well as trustworthiness of data providers in the IoS need to be faced. On the positive side, the huge number of data sources and data items may allow for data fusion, correlation and consistency checking and thus improve predictions.

- The challenges towards online quality prediction will significantly be amplified in the presence of noisy and uncertain data. This challenge has been initially addressed by the Complex Event Processing (CEP) community. Several works dealt with uncertain events, such as the effect of uncertainty in input events [31]. However, they only consider a reactive analysis of events. This means we still need to understand what the impact of uncertainty is on the prediction of future situations.
- The high dynamics of Future Internet applications requires (near) real-time processing of large data streams from distributed event sources in order to observe problems and deviations (e.g., such that adaptations can be executed). This means that current solutions, which often work on post-mortem data, need to be significantly enriched to handle such dynamics in an online fashion (see Section II-C2).

B. Assessing the Relevance of Predictions

1) *Accuracy Assessment of Online Predictions:* Ensuring that online quality prediction is accurate is critical [32]. Otherwise, wrong predictions may lead to the execution of unnecessary adaptations (in the case of false positive predictions) or missed adaptation opportunities (in the case of false negatives). As an example, unnecessary adaptations may introduce severe problems; e.g., if a working service wrongly is replaced by one with actual bugs.

Providing accurate quality predictions becomes extremely challenging in the setting of Future Internet applications, especially if they consist of third-party IoS and IoT services and thus exhibit a certain heterogeneity and dynamics (also see Section II-A2). The observed quality and functionality of those third-party services can significantly vary between different service invocations. For instance, the performance of a third-party service might depend on the load of the infrastructure at the provider’s side or the network latency if services are offered over the Internet. As another example, a failure observed at one point in time (e.g., the unavailability of a service because of a temporary overload of the computing infrastructure at the service provider side) can disappear at a later point in time.

Traditionally, accuracy of predictions is assessed in a “post-mortem” way such as to select a matching prediction technique for a specific usage setting [10]. However, in the Future Internet those usage settings or contexts may continuously change. This means that even if high accuracy is achieved in an initial setting, accuracy might quickly decrease over time. We thus need new ways to assess the accuracy in an online fashion, such as to determine – during run-time – whether to trust the predictions and take respective adaptation actions, or even to dynamically switch between different prediction techniques.

The literature provides first indications on how such a run-time assessment of accuracy could be performed. Dinda [33] proposes computing confidence intervals together with the point predictions to take an informed decisions about the deployment of computational tasks for more traditional computer-based systems. In the setting of service-oriented systems, there are proposals to compare predictions with the actual executions during runtime and to use the prediction errors to trigger the re-training of the prediction model [11]. Other proposals propose computing the statistical confidence of predictions during run-time [6]. However, those ideas need to be significantly expanded and refined to become applicable in the Future Internet setting.

2) *Contextualization of Predictions:* In practice, many service-oriented systems will – often depending on their context – have “soft” quality requirements [3]. As an example, a user accessing service offerings through her mobile phone will most probably not expect broadband connectivity and thus high performance and responsiveness if traveling through tunnels or on an airplane.

This means that the decision whether quality requirements will be violated is not a clear cut one. It may rely on objective functions; e.g, ones introduced from the point of view of the service provider [3].

In addition, utility functions that assess the “severity” of the violation from the point of view of the end-user need to be taken into account for assessing whether a predicted quality deviation in fact indicates a critical event. This means that the notion of Quality of Experience (QoE [34]) needs to be considered during prediction.

Finally, to take informed adaptation decisions, cost models become important in order to balance the costs of not taking an adaptation versus the cost of doing so (cf. [35]). As an example, the cost of the predicted violation might be smaller than the penalty to be paid for an SLA violation. Such a critical event could then be safely ignored. This means that we need to be able to quantify the risk involved in both of these options in order to take informed decisions.

C. Impact of Cloud Computing and Large Event Streams on Prediction Techniques

1) *Prediction Considering the Cloud:* For traditional networks and computer-based systems, quality prediction is usually done through continuous monitoring, followed by data analysis, mining, and modeling of the performance trends. For cloud services, however, quality prediction will be much more complicated because of the reasons we discuss below.

First, different service level agreements (SLAs) may be associated with different service subscriptions (from different service users) for the same service. This implies that simply correlating and mining the overall performance data from the cloud infrastructure may not reflect the actual situation of the service quality provisioning, as this depends

on the specific SLA (per user). Currently, infrastructure monitoring is usually done at the server level and not at the instance level, because of the relatively high overhead involved in the latter.

Second, due to the composite nature of cloud services, simply monitoring and modeling the performance of their front-end servers might not give a full, comprehensive view on what can possibly be provided by the cloud services. What we need to know are the SLAs along the chain of all the cloud services involved, as well as their real-time performance data. For example, in order to understand in detail the performance of a given software service, what we need to know is not only the performance of the front-end service server, but more importantly, the performance of the (possibly third-party) back-end infrastructure service that provides capabilities to the front-end as well as the SLAs between them. Those SLAs may include the maximum capability that can be allocated from the infrastructure service to the software service, the current performance situation of the front-end server and the back-end infrastructure service, and also contract terms, such as the price that the software service provider is willing to pay for the purchase of quality from the infrastructure service provider.

Third, for traditional software services, the quality provisioning is usually done by allocating internally available resources (computing, network, storage) and deploying software to those resources accordingly. However, for cloud services the quality provisioning needs to be done involving quality “purchases” from infrastructure service providers, i.e., resources are “purchased” on-demand. As a result, the delay incurred in the quality “purchase” needs to be taken into consideration for cloud services. Note that this delay can be much longer than the time taken for the internally available resources to be allocated in the setting of more traditional software services. This poses additional challenges to quality prediction. For instance, the lead time from when the prediction results are available until when the actual failure occurs needs to be longer, in order to facilitate the additional time required for quality “purchase” in case an adaptation is required.

2) *Leveraging Large Event Streams Towards Prediction:* The Future Internet will lead to a proliferation of heterogeneous data sources and an increased amount and frequency of monitoring events. A major source for those events will be the Internet of Things, providing access to smart sensors and devices at an unprecedented scale. On the one hand, such data may provide better prediction techniques, as more data is available to reason on future situations (see Section II-A2 above). On the other hand, such complex streams of events lead to issues such as how to process those data streams in real-time in order to perform useful online quality prediction.

The area of Complex Event Processing (CEP) is dealing with the real-time identification of meaningful situations from streams of raw events. Raw events correspond to simple

observations (e.g., a sensor read, result of pinging a server, etc.), whereas a situation (or a complex event) corresponds to some combination of raw events, such as a sequence, a trend, or some aggregation of values.

Event patterns that frequently occur in historic data can often be identified through data mining methods. Those methods can also serve towards discovery of predictive event patterns [36]. When a predictive event pattern is matched during run-time, it may predict – with some calculated confidence – the occurrence of a future event. Predictive CEP therefore corresponds to the ability to predict a future raw or complex event by real-time processing of streams of raw events. Note that both CEP and predictive CEP require that the specific event patterns be identified during design time, and that the connection between those patterns and the current or future complex events be represented in the run-time engine as a set of rules. This may pose some limitations for the applicability in the highly dynamic setting of the Future Internet and thus needs further investigation.

Predictive CEP can be potentially applied to service adaptation. A service is constantly monitored for various indicators. If the deterioration of quality of the service can be represented as a pattern on those indicators, or the data can point out the connection between the change of indicator values and the deterioration of quality, then predictive CEP is an appropriate tool to represent this connection and translate it to efficient runtime execution that combines monitoring and prediction.

Furthermore, CEP lends itself easily to expression of temporal conditions. Often, the decision making which is triggered by a prediction is time-dependent. For example, if a service is expected to deteriorate only after several hours, it may not make sense to purchase the more expensive alternative immediately. Therefore, a prediction of a future event should indicate not only which event is likely to occur, but also when it is likely to occur. If the data we have is sufficiently rich to provide an indication on the time range between the occurrence of the predictive pattern and the expected occurrence of the predicted event, then CEP rules may express this time expectation (or probability distribution over the time of occurrence of the future event), and leverage this information towards decision making.

III. CONCLUSIONS

Initial research results on online quality prediction for proactive adaptation of service-oriented systems exist. However, many research challenges in this area remain open, and many novel challenges will arise due to the emergence of the Future Internet. This paper has discussed six research challenges, within three main clusters: (i) handling heterogeneity during predictions, (ii) assessing the relevance of predictions, and (iii) impact of cloud computing and large event streams on prediction techniques. We are confident

that solving those challenges will have a profound impact on research and practice.

ACKNOWLEDGMENTS

We cordially thank Wilhelm (Willi) Hasselbring for fruitful discussions on the topic of this paper, as well as the PC members for providing very helpful suggestions for improving the paper.

The research leading to these results has received funding from the European Community's Seventh Framework Programme FP7/2007-2013 under grant agreement 215483 (S-Cube).

REFERENCES

- [1] E. Di Nitto, C. Ghezzi, A. Metzger, M. Papazoglou, and K. Pohl, "A journey to highly dynamic, self-adaptive service-based applications," *Automated Software Eng.*, vol. 15, pp. 313–341, 2008.
- [2] G. Tselentis, J. Domingue, A. Galis, A. Gavras, and D. Hausheer, *Towards the Future Internet: A European Research Perspective*. Amsterdam, The Netherlands: IOS Press, 2009.
- [3] C. Cassales Marquezan, A. Metzger, K. Pohl, V. Engen, M. Boniface, S. C. Phillips, and Z. Zlatev, "Adaptive future internet applications: Opportunities and challenges for adaptive web services technology," in *Adaptive Web Services for Modular and Reusable Software Development: Tactics and Solution*, G. Ortiz and J. Cubo, Eds. IGI Global, 2012 (to be published).
- [4] A. Metzger and C. Cassales Marquezan, "Future Internet Apps: The next wave of adaptive service-oriented systems?" in *ServiceWave 2011, Research Track*, ser. LNCS, A. Zisman, I. Llorente, M. Surridge, W. Abramowicz, and J. Vayssière, Eds. Springer, 2011.
- [5] J. Hielscher, R. Kazhamiakin, A. Metzger, and M. Pistore, "A framework for proactive self-adaptation of service-based applications based on online testing," in *ServiceWave 2008*, ser. LNCS, vol. 5377. Springer, 2008, pp. 122–133.
- [6] A. Metzger, O. Sammodi, K. Pohl, and M. Rzepka, "Towards pro-active adaptation with confidence: Augmenting service monitoring with online testing," in *SEAMS@ICSE 2010*. ACM, 2010.
- [7] O. Sammodi, A. Metzger, X. Franch, M. Oriol, J. Marco, and K. Pohl, "Usage-based online testing for proactive adaptation of service-based applications (short)," in *COMPSAC 2011 – The Computed World: Software Beyond the Digital Society*. IEEE Computer Society, 2011.
- [8] S. S. Yau and H. G. An, "Software engineering meets services and cloud computing," *IEEE Computer*, vol. 44, no. 10, pp. 47–53, 2011.
- [9] W.-T. Tsai, X. Zhou, Y. Chen, and X. Bai, "On testing and evaluating service-oriented software," *IEEE Computer*, vol. 41, no. 8, pp. 40–46, 2008.

- [10] F. Salfner, M. Lenk, and M. Malek, "A survey of online failure prediction methods," *ACM Comput. Surv.*, vol. 42, no. 3, 2010.
- [11] P. Leitner, A. Michlmayr, F. Rosenberg, and S. Dustdar, "Monitoring, prediction and prevention of SLA violations in composite services," in *IEEE International Conference on Web Services (ICWS) Industry and Applications Track*, 2010.
- [12] J. Ejarque, A. Micsik, R. Sirvent, P. Pallinger, L. Kovacs, and R. Badia, "Semantic resource allocation with historical data based predictions," in *CLOUD COMPUTING 2010, The First International Conference on Cloud Computing, GRIDS, and Virtualization*, 2010, pp. 104–109.
- [13] S. Haykin, *Neural Networks and Learning Machines: A Comprehensive Foundation, 3rd Edition*. Prentice Hall, 2008.
- [14] J. R. Quinlan, "Induction of decision trees," *Machine Learning*, vol. 1, pp. 81–106, 1986.
- [15] A. Filieri, C. Ghezzi, and G. Tamburrelli, "Run-time efficient probabilistic model checking," in *Proceedings of the 33rd International Conference on Software Engineering, ICSE 2011, Waikiki, Honolulu, HI, USA, May 21-28, 2011*, R. N. Taylor, H. Gall, and N. Medvidovic, Eds. ACM, 2011, pp. 341–350.
- [16] E. Schmieders and A. Metzger, "Preventing performance violations of service compositions using assumption-based runtime verification," in *ServiceWave 2011, Research Track*, ser. LNCS, A. Zisman, I. Llorente, M. Surrige, W. Abramowicz, and J. Vayssière, Eds. Springer, 2011.
- [17] A. Bertolino, G. D. Angelis, and A. Polini, "(role)cast: A framework for on-line service testing," in *WEBIST 2011, Proceedings of the 7th International Conference on Web Information Systems and Technologies, Noordwijkerhout, Netherlands, 6-9 May, 2011*, J. Cordeiro and J. Filipe, Eds. SciTePress, 2011, pp. 13–18.
- [18] M. Greiler, H.-G. Gross, and A. van Deursen, "Evaluation of online testing for services: a case study," in *Proceedings of the 2nd International Workshop on Principles of Engineering Service-Oriented Systems*, ser. PESOS '10. New York, NY, USA: ACM, 2010, pp. 36–42.
- [19] D. Ivanovic, M. Carro, and M. V. Hermenegildo, "Constraint-based runtime prediction of sla violations in service orchestrations," in *Service-Oriented Computing - 9th International Conference, ICSOC 2011, Paphos, Cyprus, December 5-8, 2011 Proceedings*, ser. Lecture Notes in Computer Science, G. Kappel, Z. Maamar, and H. R. Motahari-Nezhad, Eds., vol. 7084. Springer, 2011, pp. 62–76.
- [20] K. Apt, *Principles of Constraint Programming*. Cambridge University Press, 2003.
- [21] D. Ivanovic, M. Treiber, M. Carro, and S. Dustdar, "Building dynamic models of service compositions with simulation of provision resources," in *Conceptual Modeling - ER 2010, 29th International Conference on Conceptual Modeling, Vancouver, BC, Canada, November 1-4, 2010. Proceedings*, ser. Lecture Notes in Computer Science, J. Parsons, M. Saeki, P. Shoval, C. C. Woo, and Y. Wand, Eds., vol. 6412. Springer, 2010, pp. 288–301.
- [22] Y. Jamoussi, M. Driss, J.-M. Jézéquel, and H. Ben Ghézala, "QoS assurance for service-based applications using discrete-event simulation," *IJCSI International Journal of Computer Science Issues*, vol. 7, no. 4, 2010.
- [23] M. Sampath, R. Sengupta, S. Lafortune, K. Sinnamohideen, and D. Teneketzis, "Diagnosability of discrete-event systems," *IEEE Transactions on Automatic Control*, vol. 40, no. 9, pp. 1555–1575, 1995.
- [24] A. Metzger, K. Pohl, M. Papazoglou, E. Di Nitto, A. Marconi, and D. Karastoyanova, "Research challenges on adaptive software and services in the future internet: Towards an s-cube research roadmap," in *Proceedings of the ICSE 2012 Workshop on European Software Services and Systems Research – Results and Challenges (S-Cube)*, 2012.
- [25] R. Kazhamiakin, S. Benbernou, L. Baresi, P. Plebani, M. Uhlig, and O. Barais, "Adaptation of service-based systems," in *Service Research Challenges and Solutions for the Future Internet: S-Cube – Towards Mechanisms and Methods for Engineering, Managing, and Adapting Service-Based Systems*, M. Papazoglou, K. Pohl, M. Parkin, and A. Metzger, Eds. Springer, LNCS 6500, 2010, pp. 117–156.
- [26] S. Rinderle, B. Weber, M. Reichert, and W. Wild, "Integrating process learning and process evolution - a semantics based approach," in *Business Process Management, 3rd International Conference, BPM 2005, Nancy, France, September 5-8, 2005, Proceedings*, W. M. P. van der Aalst, B. Benatallah, F. Casati, and F. Curbera, Eds., vol. 3649, 2005, pp. 252–267.
- [27] C. Li, M. Reichert, and A. Wombacher, "Discovering reference models by mining process variants using a heuristic approach," in *Business Process Management, 7th International Conference, BPM 2009, Ulm, Germany, September 8-10, 2009. Proceedings*, ser. Lecture Notes in Computer Science, U. Dayal, J. Eder, J. Koehler, and H. A. Reijers, Eds., vol. 5701. Springer, 2009, pp. 344–362.
- [28] K. Ploesser, M. Peleg, P. Soffer, M. Rosemann, and J. Recker, "Learning from context to improve business processes," *BP-Trends*, vol. 6, pp. 1–7, 2009.
- [29] A. Bucchiarone, A. Marconi, M. Pistore, and A. Sirbu, "A context-aware framework for business processes evolution," in *Workshops Proceedings of the 15th IEEE International Enterprise Distributed Object Computing Conference, EDOCW 2011, Helsinki, Finland, August 29 - September 2, 2011*. IEEE Computer Society, 2011, pp. 146–154.
- [30] R. Wang and D. Strong, "Beyond accuracy: What data quality means to data consumers," *Journal of Management Information Systems*, vol. 12, no. 4, 1996.
- [31] S. Wasserkrug, A. Gal, O. Etzion, and Y. Turchin, "Efficient processing of uncertain events in rule-based systems," *IEEE Transactions on Knowledge and Data Engineering*, 2010.
- [32] A. Metzger, "Towards accurate failure prediction for the proactive adaptation of service-oriented systems (invited)," in *Proceedings Workshop on Assurances for Self-Adaptive Systems (ASAS), collocated with ESEC 2011*, 2011.

- [33] P. A. Dinda, "Online prediction of the running time of tasks," *Cluster Computing*, vol. 5, no. 3, pp. 225–236, 2002.
- [34] A. van Moorsel, "Metrics for the Internet Age: Quality of Experience and Quality of Business," in *Fifth International Workshop on Performability Modeling of Computer and Communication Systems, Arbeitsberichte des Instituts für Informatik, Universität Erlangen-Nürnberg, Germany*, 2001.
- [35] P. Leitner, W. Hummer, and S. Dustdar, "Cost-based optimization of service compositions," *IEEE Transactions on Services Computing (TSC)*, 2012.
- [36] Y. Engel and O. Etzion, "Towards proactive event-driven computing," in *Proceedings of the 5th ACM international conference on Distributed event-based system*, 2011, pp. 125–136.