# Towards Critical Event Monitoring, Detection and Prediction for Self-adaptive Future Internet Applications

Andreas Metzger[1], Michael Boniface[2], Vegard Engen[2], Stephen C. Phillips[2] and Zlatko Zlatev[2]

[1]Paluno (The Ruhr Institute for Software Technology) University of Duisburg-Essen, Essen, Germany
{andreas.metzger@paluno.uni-due.de}
[2]IT Innovation Centre, University of Southampton, Southampton, U.K.
{mjb, ve, scp, zdz@it-innovation.soton.ac.uk}

**Abstract.** The Future Internet (FI) will be composed of a multitude of diverse types of services that offer flexible, remote access to software features, content, computing resources, and middleware solutions through different cloud delivery models, such as IaaS, PaaS and SaaS. Ultimately, this means that loosely-coupled Internet services will form a comprehensive base for developing value-added applications in an agile way. Unlike traditional application development, which uses computing resources and software components under local administrative control, FI applications will thus strongly depend on third-party services. To maintain their quality of service, those applications therefore need to dynamically and autonomously adapt to an unprecedented level of changes that may occur during runtime. In this paper, we present our recent experiences on monitoring, detection, and prediction of critical events for both software services and multimedia applications. Based on these findings we introduce potential directions for future research on self-adaptive FI applications, bringing together those research directions.

## 1 Introduction and Motivation

The Future Internet (FI) will be composed of a multitude of diverse services that offer flexible, remote access to software features, physical computing resources, communication infrastructure, and middleware solutions. Emerging cloud computing technologies enable a further, drastic shift for Internet-based computing, where computing resources and software applications are provisioned on demand. Ultimately, this means that loosely-coupled Internet services will form a comprehensive base for developing value-added applications. Unlike traditional application development, which uses computing resources and software components under local administrative control, FI applications will thus heavily depend on third-party services. Those applications therefore need to dynamically adapt to an unprecedented level of changes that can occur during the runtime of those systems to be able to maintain their QoS.

Different solutions for dynamic adaptation have been developed for various areas of the FI, such as software services (IoS), as well as data and media (IoC). This paper will present those solutions and will suggest areas for future research to augment and integrate them towards solutions for self-adaptive FI applications.

After a short discussion of adaptation concepts in Section 2, the paper presents outcomes and application scenarios of three major EU projects, addressing multimedia applications (Section 3), as well as service-oriented applications (Section 4). Analysing those results, Section 5 discusses future research challenges.

## 2 Self-Adaptive Future Internet Applications

Self-adaptive systems automatically and dynamically adapt to changing conditions. The aim of self-adaptation is to reduce the need of human intervention as far as possible, thereby enabling timely responses to critical events. In order to realize self-adaptive behaviour, methods and tools that realize control loops are established to collect details from the application and its context (e.g., by exploiting monitoring mechanisms) and decide and act accordingly [14]. Two major types of control loops can be employed for application adaptation: reactive and proactive.

A *reactive* adaptation cycle performs compensation actions for a critical event that has already occurred in the system and was experienced by the user/s for some time. In a *proactive* adaptation cycle, in contrast to a reactive one, the occurrence of critical events is anticipated using prediction methods and pre-emptive system adaptation actions are undertaken. Proactive adaptation may allow the system to modify parts not yet executed, thus eliminating the need for repair and compensation.

## 3 Adaptation of Interactive Multimedia Applications

The SaaS paradigm gives the capability to application providers to reach an increasingly large number of users and also enables users to use the applications as a utility rather than having to unnecessarily invest in infrastructure and software. This is especially relevant to multimedia applications, which are typically highly interactive and especially challenging to provision so that their performance is stable and the user experience is adequate. These challenges are due to the complexities in estimating the software performance on a given hardware infrastructure and also due to the varied behaviour of the users resulting in varied and difficult to assess workloads. These challenges can be addressed by the means of adaptive environments used to operate the applications provided. Research in the IRMOS [7] and BonFIRE [3] projects address these challenges, which are discussed further below.

### 3.1 An IoC Scenario: Interactive Real-time Application as a Service

The IRMOS project addresses the challenges encountered by SaaS providers in providing soft real-time multimedia applications with guaranteed (probabilistic) QoS. Examples of such applications include interactive and collaborative film post-production, virtual and augmented reality within the engineering design process, and interactive online eLearning environments. Consider the post-production scenario, for example, in which resources are needed on short notice to host a session with many

users located in different countries to work on a film. Compute, storage and networking resources need to be selected to ensure the QoS to the users, who will be streamed a video and will interact with it by, for example, pausing, rewinding, or editing frames. One Key Performance Indicator (KPI) of such an application is whether frames are dropped, which should have certain QoS guarantees.

### 3.2 The IRMOS & BonFIRE Solutions

**QoS-Oriented Service Engineering:** The IRMOS project has developed a toolbox of techniques that allow applications with soft real-time requirements to be planned and executed on virtualised service oriented infrastructure operated by third-party service providers. In the case of the IoC scenario, there is a need for well-defined and managed Service Level Agreements (SLAs) that have guaranteed QoS. The IRMOS toolbox provides an adaptive environment of tools for negotiating, monitoring and managing SLAs and applications.

One of the main components in the IRMOS toolbox is the Performance Estimation Service (PES), which encapsulates a methodology for the SaaS provisioning planning. The PES is used for planning the deployment of the SaaS application in terms of resources that need to be reserved, as well as during the operation of the application to adapt its provisioning in response to critical events. A critical event could be, for example, observing a KPI deterioration below the agreed level, or observing a deviation from the expected application workload that may compromise the agreed QoS.

During SLA negotiation, the PES estimates the required resources for a particular application based on the predicted performance of an application model on some given resources. Details of an example model for the e-Learning application mentioned above can be found in [9]. The predicted performance is evaluated against the QoS terms set in the SLA with an objective function that encapsulates the business objectives of the SaaS provider. This is typically a function that evaluates the maximum profit for the SaaS provider, based on the expected income for running the application, minus the infrastructure costs and any penalties for not fulfilling the QoS. The IRMOS toolbox includes global and local optimisation algorithms to determine the best resources according to the defined objective function, supported by a framework of caching execution results of application models and objective functions to speed up this process.

Once the SLA is agreed, the application is deployed and run. To guarantee the agreed QoS, a performance feedback loop is implemented that facilitates reactive and proactive types of application provisioning adaptation via a Performance Feedback Service (PFS). During the application operation, KPI metrics are logged and fed back to the PFS. These observed metrics are compared against the agreed ones and in the events of deviation the resources are scaled according to predefined rules, e.g., request a 10% increase in the virtual CPU speed and RAM.

**QoS-Oriented Service Engineering for Federated Clouds:** The work on predicting application performance in IRMOS has been continued in the BonFIRE project [3], which offers a multi-site testbed of heterogeneous cloud resources for experimentation on the Future Internet. One of the main foci of the work continued in BonFIRE is

to address the challenges of predicting application performance based on the descriptions of resources offered by IaaS providers. To date, IaaS providers describe their resources in different ways, such as 'small', 'medium', 'large', or Amazon EC2 Compute Units (ECU), which do not necessarily provide very much information to end users, let alone sufficient to use as a basis for predicting application performance.

IaaS resources should ideally be described in a uniform and descriptive manner, which can be fruitful for predicting application performance. The Dwarf taxonomy introduced in [4] is one alternative, which currently comprises 13 classes of computational benchmarks [2]. A Dwarf benchmark is defined as *"an algorithmic method that captures a pattern of computation and communication"* [1]. Ultimately we could imagine each IaaS provider describing the performance of their resources in terms of a standard set of benchmark scores (such as Dwarfs) and even agreeing SLAs in those terms. Alternatively, a PaaS provider may measure the performance of many IaaS providers, adding to one of many possible services that could be offered.

Initial findings in [13] indicates that scores on Dwarf benchmarks shows promise as a means of describing computing resources in the cloud, demonstrating that they can discriminate between different compute resources, even when the IaaS provider labels them the same. Furthermore, the initial findings in [13] show different Dwarfs correlate more strongly with different applications, indicating that they can be useful in predicting application performance.

## 4 Adaptation of Service-oriented Applications

To date, the major work on adaptation of software services and service-oriented application has been centred around reactive adaptation capabilities based on monitoring [1]. As explained in Section 2, this means that adaptation is performed after a deviation or critical change has occurred. For service-oriented systems, such a reactive adaptation, however, has some important shortcomings (cf. [17] and [15]). For instance, it can take time before problems in a service-oriented system leads to monitoring events that ultimately trigger the required adaptation. Thus, monitoring events might arrive so late that an adaptation of the application is not possible anymore.

To address those challenges, the S-Cube Network of Excellence started to bring together a variety of research communities. Specifically, S-Cube members have started to devise proactive adaptation techniques based on quality prediction.

### 4.1 An IoS Scenario: The Olympic Games App

A possible example for a service-oriented system which heavily relies on third-party services is shown in Figure 3. It depicts an "Olympic Games App", which is intended to support end-users in gathering access to information about specific sporting events during the London 2012 Olympic Games. The third-party services used are publically available services (including ones presented in [4]).

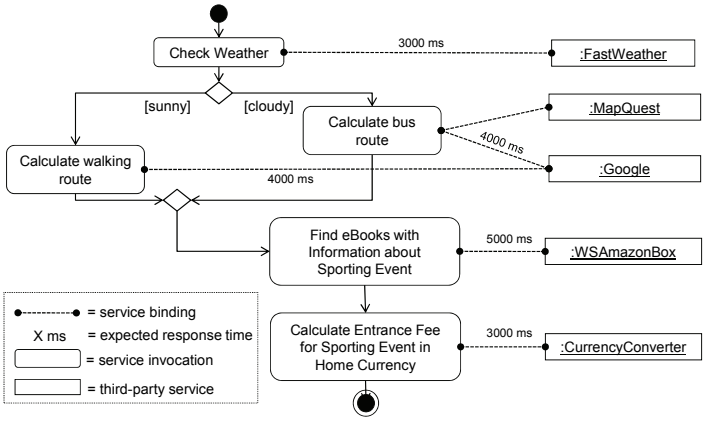


**Figure 1. Workflow and Service Composition of Olympic Games App**

One QoS requirement towards the application is that it should provide its response within 15 seconds. As the App relies on third party services, it might happen during run-time that the MapQuest service replies slower than expected (say it responds in 4500 ms instead of 4000 ms). Thus, it may be better to adapt the App to use the Google service instead. Of course, once performance has been violated, such an adaptation would come too late. Thus, such modifications need to be done proactively.

### 4.2 The S-Cube Solutions

From the different, complementary, solutions to forecast problems that have been developed in S-Cube [15] this paper focuses on two representative solutions[1].

**Prediction of Service Failures:** Key to proactive adaptation is the ability to predict the future quality of a service-oriented system's constituent services. Typically, monitoring is used to assess the quality of a service during its operation. Based on monitoring data, failures are predicted and thus the need for adaptations is identified. However, due to its observational (passive) nature, monitoring may not ensure a comprehensive coverage of all relevant service executions.

As a solution, we propose to employ online testing of a service-oriented system's constituent services as a means to produce more frequent data points. During online testing, constituent services are systematically tested in parallel to the normal use and operation of the service-based system. Specifically, we introduced a novel approach for online testing, which performs an inverse usage-based test of the services [15]. This means that if a service has seldom been used in a given time period, and thus not enough monitoring data has been collected, dedicated tests are performed.

Initial empirical results [10] indicate that the accuracy of failure prediction using the complementary use of testing and monitoring is better than the prediction using monitoring data only. Figure 4 shows these results for the Google service from the

[1] A full list and description of these techniques is accessible at http://www.s-cube-network.eu/QP

above scenario, indicating the capability of the prediction techniques in avoiding unnecessary adaptations and not missing required adaptations. The diagram shows a comparison with two baseline techniques building on monitoring.

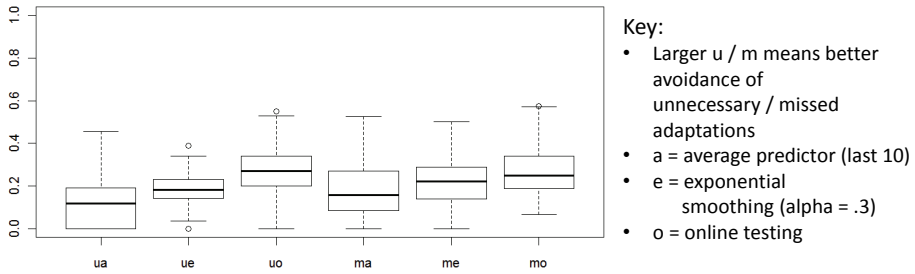


**Figure 2. Improved prediction through online testing (Google example)**

**Prediction of SLA Violations**: Even though a constituent service may not have met its expectations, this does not necessarily imply that the service-oriented application will violate its QoS expectations (the 15 seconds in our example). Although the MapQuest service might have responded slower than expected, the preceding Fast-Weather service might have been much faster than expected, thereby compensating for the slower response of MapQuest.

To predict whether deviations from a service's expected QoS have an impact on the SLA of the service-oriented application, in [17] we thus proposed employing run-time verification (model-checking) techniques which take into account a model of the workflow as well as evidence about the execution of services along the workflow. Initial experimental results indicate that the approach may considerably reduce the number of unnecessary adaptations to a level of as small as 12%.

# 5 Towards Monitoring and Adaptation of FI Applications

## 5.1 A Future Internet Scenario: Transport and Logistics

Transport and logistics applications currently face critical obstacles in achieving more reliable, lower cost and environmentally friendly transport. Future Internet technologies can facilitate radical improvements not only for optimizing existing processes in international transport and logistics, but also for completely new and innovative logistics processes. Specifically, the operational procedures of the involved stakeholders could be significantly improved and intermediate steps currently needed due to lack of control and visibility could be rationalized [15].

A key capability in such a setting will be predictive, distributed event monitoring and detection which will allow for an efficient treatment of delays and other unforeseen events by proactively adapting the transport and logistics processes. As shown in Figure 5, Future Internet technologies for real-time integration of information from sensor networks and smart infrastructures (IoT), live video data capture and streaming (IoC), together with access to those data sources any time and in any place (through

NoF) allow supply chain partners (offering their capabilities through the Internet of Services) to obtain an extended information base.

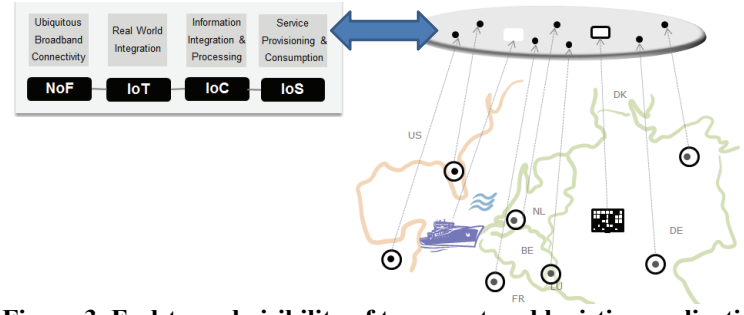


**Figure 3. End-to-end visibility of transport and logistics applications [11]**

### 5.2 Requirements towards Future Solutions

One key enabler for adaptive FI applications, such as the one above, will be the ability to have a seamless and consistent way of monitoring, detecting and predicting critical events for the different areas of the Future Internet and thus for the types of services offered. As an example, assuming that an IoT-based RFID-sensor that would track packages within a warehouse fails, seamless monitoring would allow switching to video data and video analysis to track those packages.

Further, due to the very large scale of FI applications, this requires significant progress towards decentralized and highly dispersed facilities for monitoring, detecting and predicting critical events. On the other hand, the high number of data from different sources (such as services, things or media items) may provide refined approaches towards detecting and predicting critical events; e.g., by correlating trends from those various sources or applying more powerful complex event processing facilities.

### 5.3 Detection

As indicated in Section 3.1, many practical applications will have some "soft" QoS requirements. This means that the decision whether QoS expectations have been violated is not a clear cut one, but needs to rely on objective functions (such as the one introduced from the point of view of the provider in IRMOS). 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 monitoring event indicates a critical event. This means that the notion of Quality of Experience (QoE) needs to be considered during detection. When talking about QoE, the role of the context in which an application executes will have an additional impact; for instance, a user travelling in an airplane might not necessarily expect to have broadband access to his services and thus would be satisfied with much less connectivity.

### 5.4 Prediction

As discussed in Section 4.2, the prediction of critical events (such as deviations in the QoS of constituent services or SLA violations of service-oriented systems or multimedia applications) will always have a margin for error. Thus, in order to decide whether a proactive adaptation decision should be based on the prediction of a critical event, metrics and tools need to be in place in order to assess the accuracy of such a prediction. As visible from the empirical results of S-Cube (e.g., see Section 4.2), there is still major room for improvement. Similar to what has been mentioned in the introduction to this section, we speculate that the availability of a wider range of data sources in the Future Internet will provide an added level of accuracy, as prediction can be based on more and more frequent information. However, how to exploit this in an integrated and coordinated way is still an open question.

In addition, the highly dynamic setting of FI applications will make it difficult to select the best and most suitable prediction technique during design time. As discussed in Section 3.2, benchmarking application behaviour to better predict infrastructure properties is quite challenging particularly since there is currently no uniform way of describing resource offerings in the cloud. This is a challenge that is currently investigated in the BonFIRE project, but is only one piece of the puzzle for achieving scalable and dynamic prediction of application performance. For example, dynamic approaches to adapting the prediction models based on operative data could be an interesting research direction. Moreover, taking into account an applications workload is still an open question, as discussed in [13].

Finally, similar to the discussion on how to assess critical events in Section 5.3, predicted problems may be "contextualized" in order to better understand the criticality of the forecasted event. Here, again QoE considerations may play an important role. In addition, cost models become important in order to balance the costs of not taking an adaptation vs. the cost of doing so and thereby being able to quantify the risk involved in both these decisions. 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.

## 6 Conclusions

This paper has discussed recent research on monitoring, detection, and prediction of critical events for both software services and multimedia applications. Based on these findings potential directions for future research on self-adaptive Future Internet (FI) applications have been discussed. To enable adaptive FI applications, it is clear that a seamless and consistent way of monitoring, detecting and predicting critical events is required. Besides other challenges, applications need to be able to dynamically adapt to an unprecedented level of changes that can occur during runtime.

**Acknowledgements:** The research leading to these results has received funding from the European Community's Seventh Framework Programme FP7/2007-2013 under grant agreements 214777 (IRMOS), 257386 (BonFIRE) and 215483 (S-Cube).

We thank Eric Schmieders, Osama Sammodi and Clarissa Marquezan. Their original contributions have provided a valuable baseline for this paper.

## References

[1] K. Asanovic, R. Bodik, B. C. Catanzaro, J. J. Gebis, P. Husbands, K. Keutzer, D. A. Patterson, W. L. Plishker, J. Shalf, S. W. Williams, and K. A. Yelick, "The Landscape of Parallel Computing Research: A View from Berkeley," Electrical Engineering and Computer Sciences, University of California at Berkeley UCB/EECS-2006-183, 2006.

[2] K. Asanovic, R. Bodik, J. Demmel, T. Keaveny, K. Keutzer, J. Kubiatowicz, N. Morgan, D. Patterson, K. Sen, J. Wawrzynek, D. Wessel, and K. Yelick, "A View of the Parallel Computing Landscape," Communications of the ACM, vol. 52, Oct 2009.

[3] BonFIRE. (2011). EC FP7-ICT BonFIRE Project. Available online: http://www.bonfire-project.eu/

[4] B. Cavallo, M. Di Penta, and G. Canfora. An empirical comparison of methods to support QoS-aware service selection. In PESOS@ICSE'10, New York, NY, 2010. ACM.

[5] P. Colella, "Defining Software Requirements for Scientific Computing," DARPA HPCS Presentation, 2004.

[6] E. Di Nitto, C. Ghezzi, A. Metzger, M. Papazoglou, K. Pohl. A journey to highly dynamic, self-adaptive service-based applications, Autom. Softw. Eng., 15(3-4), 2008.

[7] R. Franklin, A. Metzger, M. Stollberg et al. "Future Internet technology for the future of transport and logistics (invited)," in ServiceWave 2011, Future Internet PPP Track, ser. LNCS. Springer, 2011.

[8] IRMOS (2011). EC FP7-ICT IRMOS Project. Available online: http://www.irmosproject.eu/Service_Management.aspx

[9] T. Cucinotta, et al., 2010. Virtualised e-Learning with real-time guarantees on the IRMOS platform. In Proceedings of SOCA 2010: 1-8

[10] 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.

[11] A. Metzger and C. Cassales Marquezan, "Future Internet Apps: The next wave of adaptive service-oriented systems?" in ServiceWave 2011, Research Track, ser. LNCS. Springer, 2011.

[12] M. Papazoglou, K. Pohl, M. Parkin, A. Metzger (Eds.) Service Research Challenges and Solutions for the Future Internet: S-Cube – Towards Mechanisms and Methods for Engineering, Managing, and Adapting Service-Based Systems. Springer, 2010.

[13] S. Phillips, V. Engen, J. Papay. Snow White Clouds and the Seven Dwarfs. Submitted to IEEE CloudCom 2011.

[14] F. Salfner, M. Lenk, and M. Malek. A survey of online failure prediction methods. ACM Comput. Surv., 42(3), 2010.

[15] O. Sammodi, A. Metzger, X. Franch, M. Oriol, J. Marco, and K. Pohl, "Usage-based online testing for proactive adaptation of service-based applications," in COMPSAC 2011 – The Computed World: Software Beyond the Digital Society. IEEE Computer Society, 2011.

[16] S-Cube (2011). EC FP7-ICT S-Cube Project. Available online: http://www.s-cube-network.eu/

[17] E. Schmieders and A. Metzger, "Preventing performance violations of service compositions using assumption-based run-time verification," in ServiceWave 2011, Research Track, ser. LNCS, A. Zisman, I. Llorente, M. Surridge, W. Abramowicz, and J. Vayssiere, Eds. Springer, 2011.