J Internet Serv Appl (2012) 3:1–4
DOI 10.1007/s13174-011-0058-3

GUEST EDITORIAL

Guest editorial: Special issue on the future of middleware

Valérie Issarny · Gordon Blair

Published online: 27 January 2012

1 Introduction

We are witnessing a period of rapid and intense change in distributed systems, at a rate that is unprecedented since the inception of the subject in the early 1980s. With the advent of cloud computing, for example, we can see the deployment of very large-scale distributed systems offering a range of novel and exciting new services, including interesting new paradigms for large-scale computation. Distributed systems are also becoming significantly more heterogeneous spanning very small devices embedded in the physical environment around us through data centres housing massive cluster computers. In addition, users of distributed systems are often on the move resulting in significant context changes over time, which the system must adapt to. Networking technologies also continue to evolve with, for example, the emergence of a range of new ad hoc networking techniques and peer-to-peer approaches to implementing core network services.

Middleware retains a core role in distributed systems, offering key properties such as abstraction, interoperability, openness, and supporting a range of non-functional properties. However, middleware is coming under increasing pressure given the trends highlighted above:

- What are the right abstractions for the development of future distributed systems given the scale of complexity of the underlying infrastructure? How can we abstract over this complexity? What do we need in terms of middleware APIs, programming languages, and associated software engineering methodologies?
- How do we achieve interoperability and openness in this new world we find ourselves in, especially given the extreme heterogeneity we encounter in the distributed systems of today? What principles and approaches do we need to deal with such extreme heterogeneity? Do existing approaches to interoperability and openness still work?
- How do we achieve the desired level of dependability and security? Again, what principles and techniques do we need to achieve such non-functional properties and how do we embed such techniques within the architectures of our middleware platforms?

These are very difficult questions to answer and it is clear that the middleware community faces some major challenges over the next few years in keeping up with the pace of change. This special issue captures discussions on the future of middleware as presented at the inaugural Future of Middleware event (FOME'11) that was partly supported by the EC FET IP CONNECT project (http://connect-forever.eu/). The event was held in Lisbon on 13th December 2011 in conjunction with the ACM/IFIP/USENIX Middleware conference, the premier conference in the area of middleware principles, architecture, and engineering (http://2011.middleware-conference.org/fome/fome2011).

This event brought together a number of invited leading researchers (see picture below) in the field, selected to offer comprehensive coverage of the key issues identified above. The aim of the event was to take stock of where we are in middleware, to address the key challenges facing the field, to establish an agenda for the next wave of middleware re-

V. Issarny
Inria Paris—Rocquencourt, Domaine de Voluceau, B.P. 105,
78153 Le Chesnay, France
e-mail: valerie.issarny@inria.fr

G. Blair (✉)
School of Computing and Communications, InfoLab21, Lancaster
University, South Drive, Lancaster LA1 4WA, UK
e-mail: gordon@comp.lancs.ac.uk

Table 1 Papers presented at FOME'11

Topic	Paper 1	Paper 2	Paper 3
Developing highly complex distributed systems	White et al.	Autili et al.	Ghosh et al.
Highly heterogeneous and dynamic distributed systems	Mottola and Picco	Denker et al.	McKinley et al.
Very large-scale distributed systems	van Steen et al.	Cook et al.	Boutaba et al.
Security and dependability	Campbell et al.	Little et al.	Joosen et al.
Future usage of distributed systems	Capra and Quercia	Paolucci and Souville	Dearle and Dobson

search, and most importantly, to stimulate researchers and build a community to address the significant challenges we face. In a nutshell, what should middleware look like in 2020?

The papers were organised around five key topics, with three invited papers on each, as shown in Table 1. We now look at the papers under each topic in more detail.

2 Developing highly complex distributed systems

This first topic addresses one of the key underlying motivations for this workshop, the sheer complexity of many distributed systems today, examining this topic from three complementary perspectives: a middleware perspective, a software engineering perspective, and a language perspective.

The paper by White et al. discusses the emergence of complex distributed systems highlighting a number of key drivers in areas as diverse as air traffic control, smart grid systems, and integrated health care delivery. The paper presents 5 key challenges that future middleware developers must resolve: (i) encapsulating heterogeneity at scale, (ii) supporting diverse Quality of Service requirements, (iii) deriving valid, high-performance configurations of highly configurable infrastructure platforms, (iv) dynamic configuration and reconfiguration, and (v) ensuring robust extensible and adaptive programming and communication models.

Autili et al. take the argument further by suggesting that complex distributed systems not only impact on middleware developments but also require a fundamental rethink of the entire software engineering process. Like White et al., they argue that dynamism is a key feature of future systems; indeed, they see the traditional boundaries between static and dynamic activities breaking down and development support mingling with run-time support, thus invading the traditional middleware territory.

The final paper in this set examines the key issue of programming language support in such complex systems. One of the authors, Steve Vinoski, has previously been at the centre of the move toward programming-language independent middleware such as CORBA that then require a mapping between the heterogeneous programming languages used and the underlying language-neutral middleware, an approach which he now sees as flawed and error-prone. This paper by Ghosh et al. charts the increasing use of a range of programming languages including Erlang, Scala, Haskell, and Clojure that allow distributed and embedded systems programming at a higher level of abstraction. The paper concludes by speculating about the possible influences that such alternative programming languages could have on the future of middleware and distributed systems development.

3 Highly heterogeneous and dynamic distributed systems

This topic examines two key facets of the complexity we refer to above, that is the increasingly high levels of heterogeneity and dynamism in distributed systems.

The first paper by Mottola and Picco considers the developments in wireless sensor networks, one of the key drivers for the increasing levels of heterogeneity as distributed systems encompass the ubiquitous computing vision. The paper focuses on the role of middleware in wireless sensor networks noting that applications in this domain are often developed directly on top of the underlying operating systems rather than benefiting from the higher level programming abstractions and support for interoperability offered by middleware. The paper speculates about the role of middleware in wireless sensor networks and whether the middleware should be similar to that used in other environments. The paper also highlights open research challenges and potential pitfalls in this domain.

The second paper by Denker et al. examines the trend toward cyber-physical systems, that is (distributed) systems that directly interface to the physical world. Examples of such cyber-physical systems include environmental monitoring and management systems, smart grids, and healthcare systems. Mirroring the paper by Mottola and Picco, this paper addresses the roles and needs of middleware in such applications with a particular focus on how techniques such as reflection and cross-layer approaches can be used to provide an adaptive and flexible solution to achieving dependability in such complex environments.

The final paper in this area, by McKinley et al., picks up on this topic of adaptive systems and dependability, taking inspiration from the biological world and exploring how an evolutionary approach can be used in the development and run-time support of dynamically-adaptable, high assurance middleware. The authors argue that, because of the open-ended nature of evolutionary computing and the potential for unexpected and unanticipated solutions, the approach must be coupled with rigorous development tools and run-time support to ensure that the resulting systems behave in accordance with requirements. The paper concludes by highlighting open research challenges in this area.

4 Very large-scale distributed systems

This topic focuses on the third great challenge related to complex distributed systems, that of the very large-scale of many modern distributed systems (building on the dimensions of heterogeneity and adaptation as considered above).

The paper by van Steen et al. charts the traditional challenges of scalability in terms of processes, resources and users as well as crossing multiple administration domains. The paper develops the argument further by highlighting the increasing importance of respecting the needs of users in what are fundamentally socio-technical systems and then examining the implications of this statement for scalability. The paper considers three challenges in dealing with scalability in socio-technical systems, arguing that there are no easy solutions at present to these challenges. The three challenges are: the design of socio-technical systems, the subsequent long-term management of such systems, and the key-trade-offs between user privacy and providing information to the underlying infrastructure.

The next two papers focus on cloud computing, one of the key drivers for the emergence of very large-scale systems.

The paper by Cook et al. highlights the importance of manageability in cloud environments, where manageability is defined as "the collective processes of deployment, configuration, optimisation, and administration during the lifecycle of IT systems and services." The paper highlights key trends associated with manageability and also examines three specific use cases, namely enterprise computing, Web 2.0 developments, and also high performance computing (HPC), highlighting the key challenges in each area.

While many observers of cloud computing tend to emphasise the data storage elements of cloud infrastructure, the paper by Boutaba et al. rightly emphasises the exciting role of distributed computation in the cloud. The authors argue that, whereas cloud computing often assumes a certain level of homogeneity in applications and execution environments, in reality there can be high levels of heterogeneity in the nature of the underlying clusters and also in the workloads imposed by the applications. This places heavy demands on the underlying scheduling algorithms underpinning cloud computation services. The paper charts the emergence of technologies such as MapReduce and highlights the key research challenges associated with heterogeneity particularly for job scheduling. Clearly, the analyses of this paper also extend to the implications of heterogeneity for other cloud services.

5 Dependability and security

This topic is concerned with meeting dependability and security challenges in the complex distributed systems of today.

The first paper by Campbell et al. looks at the concept of assured computation in the context of large-scale, dynamic systems of systems, highlighting the importance of approaches that are both end-to-end and also cross-layered. Assured computation is a broad concern covering security, dependability, and timeliness, including protection from malicious attacks and accidental failures. The key contribution is a discussion of the architecture of middleware platforms to support assured computation (including assured cloud computing) featuring consideration of algorithms and approaches which achieve security, enhance end-to-end timeliness, detect security policy or reliability requirement violations in a given configuration, dynamically configure resources for a given workflow based on security policy and reliability requirements, and estimate the probability of completion of a workflow for a given configuration.

The second paper by Little et al. looks specifically at dependable distributed computing and its implications for middleware. The paper starts by taking a historical view of the role of middleware in supporting dependable computing highlighting the key concepts of replication and transactions and how they have influenced standards such as Java EE. As with all the papers in this special issue, they recognise the profound changes in distributed systems, particularly in the way applications are constructed from a series of underlying services in what is inevitably a global and potentially large-scale distributed infrastructure. They see customisation as a key requirement for future dependability support.

The final paper by Joosen et al. in this area picks up strongly on the themes of the first two papers seeking what they call trustworthiness (covering performance, availability, and various aspects of security) and how this should be delivered by middleware as reusable services. They also seek customisation of this underlying support, and present an initial experiment in demonstrating how custom security middleware can be generated on-the-fly in a way that is both cost effective and that supports traceability from requirements to deployed services.

6 Future usage of distributed systems

Distributed systems are not only being challenged by the increasing complexity of the underlying infrastructure but also by the new styles of applications and services now being considered.

A key illustration of this is the rapid emergence of social computing, that is supporting social behaviours in distributed systems through tools such as wikis, blogs, and social networking sites. Social computing is also now being taken a step further as the associated tools exploit the emergence of mobile devices and also start to embrace the new potential offered by ubiquitous computing. The paper by Capra and Quercia examines the emergence of social computing and considers the implications for middleware, in particular reflecting on how middleware needs to adapt to support social computing. The paper offers a framework to reason about a decade of research in social computing and highlight key research challenges and opportunities for the middleware community in this area.

The second paper by Paolucci and Souville examines the crucial problem of data interoperability in distributed systems, highlighting the real challenges in achieving interoperability in the complex distributed systems of today. Traditionally, interoperability solutions have operated at a syntactic level in seeking mappings between different data structures and data formats. This paper examines a different approach, derived from ontologies and the semantic web, to support more sophisticated conceptual reasoning about this mapping process. The paper discusses the potential of this approach, also highlighting outstanding research challenges and potential pitfalls.

The final paper in the special issue by Dearle and Dobson considers the potential of sensor-rich systems in supporting scientific experimentation, and the underlying challenges associated by the coordination of such systems to achieve the overall integrity of the associated scientific inquiry. The paper argues that this has powerful implications for not just middleware but all aspects of software development including programming language provision and formal methods support. The paper focuses specifically on the use of self-managed component-based approaches and how to achieve the desired levels of assurance in such adaptive environments. The paper draws some overall conclusions related to the integration of mission descriptions and data provenance into the overall scientific workflow in such systems.

7 Final comments

The guest editors of this special issue would like to thank the many people who have contributed to this special issue and to the associated FOME workshop. First and foremost, we acknowledge the hard work of the authors of the papers, all leading figures in the area, who through their deep knowledge of distributed systems, have produced a magnificent account of the challenges facing middleware today. We also thank the organisers of the Middleware conference in Lisbon for their support in hosting the Future of Middleware event. We particularly thank the General Chairs of the event, Paulo Ferreira and Luis Veiga, for their help with local arrangements and for ensuring everything went smoothly on the day. We also note the contributions of the audience for making FOME'11 such a vibrant and exciting event. If you missed it, we hope this special issue provides a strong record and lasting legacy of the key issues that were discussed at FOME'11. You may also refer to http://connect-forever.eu/ for presentations. Finally, we would like to thank Fabio Kon and all the staff at Springer for making the production of this special issue such an enjoyable and smooth process.