Emergent Middleware

Paul Grace, Gordon Blair, Valérie Issarny

Emergent Middleware

by Paul Grace, Gordon S. Blair and Valerie Issarny

We are moving towards a world in which everything is connected. Yet such a goal highlights the deficiencies of today's systems platforms in achieving a fundamental property of distributed systems, namely interoperability. Faced with extreme heterogeneity of computational devices and networks, how can we ensure that every system can talk to every other system?

Interoperability is the ability for two systems to exchange, understand and use each other's data, and is a long-standing problem in the field of distributed systems. However, the emergence of pervasive computing and the Internet of Things have brought about new challenges to achieving universal interoperability. Today's complex distributed systems are characterized by extreme heterogeneity and spontaneous interactions. Computational devices ranging from embedded devices, sensors, and smartphones through to cluster machines and the Cloud use a wide range of communication networks and middleware protocols to communicate with one another. However, as soon as two systems adhere to heterogeneous protocols (from application down network layers) to interact with each other, interoperability is impossible. Standards are a well-established approach to rectifying these types of problems, eg the set of CORBA standards from the OMG, and the set of Web Services standards from the W3C. Where two systems agree upon such a standard, interoperability can be guaranteed. However, systems may encounter one another spontaneously where no such agreement is possible, and hence where the communication protocols differ they cannot interoperate.

The CONNECT project, which began in February 2009 and concludes at the end of 2012, is an EU Future and Emerging Technologies project involving partners whose expertise covers middleware, software engineering, formal methods, machine learning, software synthesis and systems dependability. The aim of the project is to overcome interaction protocol heterogeneity at all layers, on-the-fly, by using a revolutionary approach that dynamically generates the necessary interoperability solution to connect two heterogeneous systems. We term this new style of middleware: *Emergent middleware*.
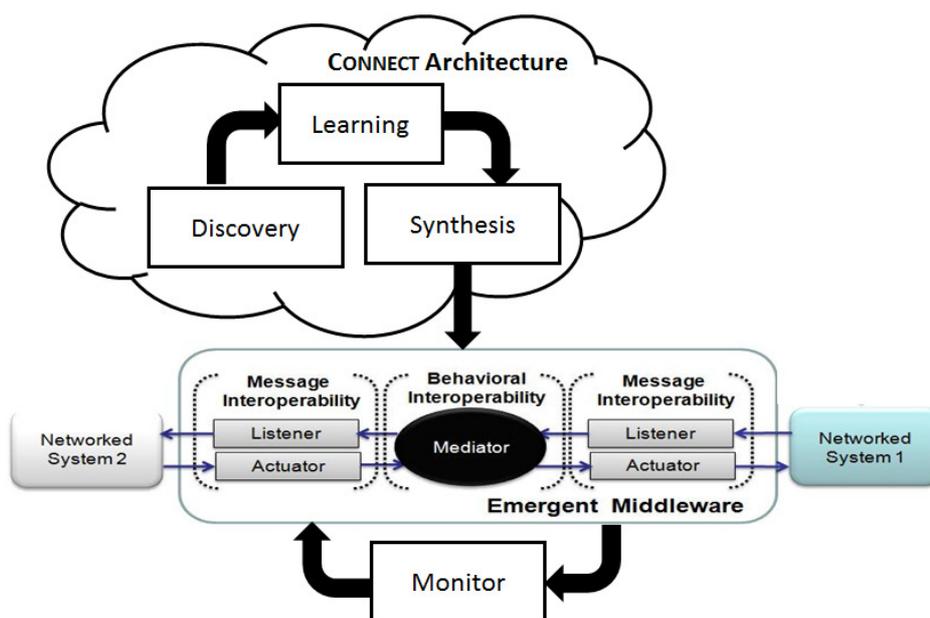
Figure 1 illustrates an emergent middleware solution. The figure shows a CONNECTor ensuring interoperation between two networked systems and performs two important functions:

1. *Message interoperability* is dedicated to the interpretation of messages from/toward networked systems (listeners parse messages and actuators compose messages); by plugging in the correct listeners and actuators we can communicate with any legacy protocol.
2. Behavioural interoperability mediates the interaction protocols run by the communicating networked systems by translating messages from one protocol to the other.

In order to dynamically generate the above solution on-the-fly, CONNECT employs a runtime architecture that executes the following phases (these phases are depicted in Figure 1):

1. *Discovery*. Systems in the local environment are discovered and their information (eg an interface description) is used to build a model of the service provided and what middleware it utilizes. Ontological information is then used to understand this provision and obtains better matching with other systems, ie does one system provide the behaviour another requires (irrespective of the heterogeneity)?
2. *Learning*. Active learning algorithms determine more precisely the behaviour of a given system in terms of the sequence of operations executed to provide a particular service and therefore enhance the model of a system.
3. *Synthesis*. The models of the two systems are used to calculate the necessary mediator that will ensure the translation between two systems. The generated mediator is used to create an emergent middleware, *aka* CONNECTor, which is deployed to achieve the interoperability between two systems.
4. *Evolution*. The deployed CONNECTor is monitored to ensure it is meeting the non-functional requirements of the connection (eg performance in terms of a minimum level of network latency). Where the requirement is not met, the architecture cycle is re-run to generate a new CONNECTor.

The above functionalities are supported by dedicated *Enablers* that are being developed as part of CONNECT and released open source from the project Web site. In particular, the Starlink open source software framework offers the tools to develop Message interoperability solutions. Currently, Starlink has been successfully used to dynamically generate CONNECTors directly between middleware protocols such as: CORBA to XML-RPC, and Service Location Protocol to Bonjour; it has also been used to build bridges between heterogeneous application systems eg a Flickr client application built using XML-RPC interoperating with the Picasa REST API.

Figure 1: On the fly generation of emergent middleware

CONNECT has illustrated the possibilities of emergent solutions. However, the inherent openness and probability of such solutions raise important challenges, especially when deployed at Internet scale. Here it must be reliably able to produce correct mediators and also be secure against malicious threats, which are other areas of active research in CONNECT.

Links:
The Connect Project website:

http://www.connect-forever.eu

The Starlink Framework:

http://starlink.sourceforge.net

Please contact:

Paul Grace
School of Computing and Communications, Lancaster University, Lancaster
E-mail: p.grace@lancaster.ac.uk

Gordon Blair
School of Computing and Communications, Lancaster University, Lancaster
E-mail: gordon@comp.lancs.ac.uk

Valerie Issarny (Project co-ordinator)
Inria Paris-Rocquencourt, France
E-mail: Valerie.Issarny@inria.fr