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.

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:
http://www.connect-forever.eu
Starlink framework:
http://starlink.sourceforge.net

Please contact:
Paul Grace
Lancaster University, Lancaster, UK
E-mail: p.grace@lancaster.ac.uk

Gordon Blair
Lancaster University, Lancaster, UK
E-mail: gordon@comp.lancs.ac.uk

Valerie Issarny
Inria, France
Arles project-team
E-mail: Valerie.Issarny@inria.fr

Never-stop Learning: Continuous Validation of Learned Models for Evolving Systems through Monitoring

by Antonia Bertolino, Antonello Calabrò, Maik Merten and Bernhard Steffen

Interoperability among the multitude of heterogeneous and evolving networked systems made available as black boxes remains a tough challenge. Learning technology is increasingly employed to extract behavioural models that form the basis for systems of systems integration. However, as networked systems evolve, their learned models need to evolve as well. This can be achieved by collecting actual interactions via monitoring and using these observations to continuously refine the learned behavioural models and, in turn, the overall system. This approach is part of the overall CONNECT approach.

In today's networked world, software is highly pervasive and increasingly existing components and services are dynamically aggregated into newer composite systems according to ever changing and demanding user expectations. The process of system integration and validation would presuppose the availability of models for the composition of software pieces. However, in real life software components that are released as black boxes do not generally provide models; this is particularly true with respect to behaviour. Learning techniques are thus being leveraged to enable the automatic derivation of formal behavioural models for black box components.

In active automata learning (see Figure 1), queries composed of a target system's input symbols are launched for execution. Based on the system responses, the learning algorithm creates and incrementally refines a hypothesis model of the system's behaviour.

However, this hypothesis is not guaranteed to be correct, which of course has implications for the correctness of the integrated system. Moreover, even if the constructed model was correct when it was built, the systems learned might subsequently evolve making the corresponding automata obsolete.

To tackle such issues, we propose a Never-stop Learning approach, where learning is not considered complete with

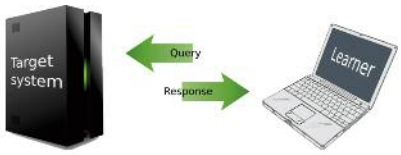

Figure 1: Active learning

the building of the model. Instead, we keep the black-box components under observation by run-time monitoring in order to capture the event flow of components' interactions within the integrated system, so that we can check whether the actual observed behaviour complies with the hypothesized model. We then attempt to retrace the sequence of monitored observations on the states of the current learned model. If this is not possible, ie there are no states in the hypothesis that can explain the observed interactions, this provides evidence of a mismatch between the learned hypothesis and the actual target systems.

The sequence of real-life observations that are not contained in the learned model can then be transformed into a counterexample for the learning algorithm, by retracing the observed steps in the hypothesis until diverging behaviour becomes apparent. This counterexample will thus trigger the construction of a refined hypothesis by the learning algorithm.

In Figure 2 the overall learning and monitoring approach is illustrated: The (publish/subscribe) monitoring infrastructure collects (through the instrumented connector) and makes sense of relevant data about the target system behaviour. The grouped observations are retrieved from the bus by the state tracker and matched against the state space of the learner's current hypothesis. If no such matching is possible, the trace of incoming observations is evidence of model mismatch,

which then is transformed into a counterexample by the evidence evaluator. Counter-examples are provided to the learning algorithm to update the model accordingly.

The approach is currently undergoing implementation within the European Future and Emerging Technologies project CONNECT, which addresses cross-platform, cross-middleware and cross-paradigm interoperability barriers. CONNECT envisions a dynamic environment populated by heterogeneous networked systems willing to communicate. The project aims at seamlessly supporting their interaction via mediating software bridges, called CONNECTors, which are synthesized and deployed on-the-fly.

The generation of these CONNECTors relies on learning technology to extract behavioural models for the networked systems which are fed to the synthesis enabler. In CONNECT, the learning is done via LearnLib, a flexible component-based framework for active automata learning, and monitoring is performed by GLIMPSE, a publish-subscribe event-based monitoring infrastructure. The synthesized CONNECTor is equipped with monitoring probes, which collect information on actual run-time system interaction. GLIMPSE then makes sense of the collected data, employing a complex event processor engine and rule-based delegation of observations. Transported over a shared message bus, the elaborated information is retrieved by the learning component in the CONNECT architecture and subsequently analysed.

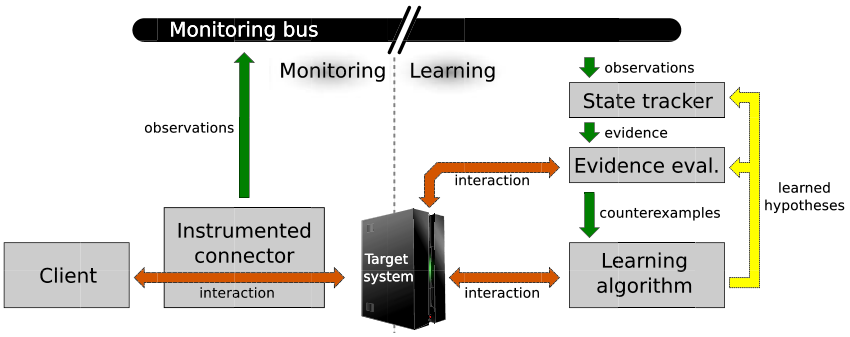

Figure 2: Never-stop Learning approach: learning and monitoring combined.

The described coupling of learning and run-time monitoring is just one example where technology is combined in novel ways to realize the CONNECT ambitious vision of eternal connection. The project, currently in its third year, will release a complex architecture populated with enablers not only for learning and monitoring, but also for discovery, synthesis, trust management, dependability and performance analysis, which altogether transparently will support interoperability among heterogeneous and evolving networked systems.

Links:
http://www.connect-forever.eu
http://labsewiki.isti.cnr.it/labse/tools/glimpse/public/main
http://www.learnlib.de

Please contact:
Antonello Calabrò
ISTI-CNR, Italy
Tel: +39 050 315 3463
E-mail: antonello.calabro@isti.cnr.it

Maik Merten
Technische Universität Dortmund, Germany
Tel: +49 231 755 7759
E-mail: maik.merten@tu-dortmund.de

PINCETTE – Validating Changes and Upgrades in Networked Software

by Pamela Farries and Ajitha Rajan

Europe relies on the availability and flawless functioning of distributed infrastructure services such as electricity, water, communication, transportation and environmental management, which are all increasingly controlled by software. Currently, however, the potential for innovation is limited owing to the inherent risks and costs of upgrades. For example, incompatibilities when simultaneously running old and new versions of control software can result in major service outages. Similarly, fast moving technology, such as sensors, could enable advances in safety critical applications such as aerospace; however revalidating avionics software to introduce new niche functionality is prohibitively expensive. The PINCETTE project has been set up specifically to ensure safe infrastructure upgrades and enable rapid product development by providing solutions for continuous validation. Certainty in the safety of upgrades will allow communities to confidently apply and leverage the efficiency gains and advantages of upgrades.

The PINCETTE project targets the problem of analyzing and validating complex systems upgrades. London's power grid, for instance, has over 4000 medium-voltage substations; the software in controllers for these substations must prevent power outages and ensure safety. A malfunctioning device can result in huge economic damage and can even threaten human lives. Upgrades to software might be needed to improve efficiency or to fix a bug, but validating this upgrade is complicated owing to the software being distributed across substations and the entire grid. Additionally, the grid cannot be switched off when validating an upgrade, making upgrades across such a network a huge safety concern.

With current technology, the only option is to supplement each incremental