To address these issues our research agenda is twofold. First, we investigate established audit, security and safety analysis methods to extract the relevant high level security properties. Safety analysis methods are typically used in the peripheral domain and security analysis methods in the backend. These need to be combined as ‘safety and security co-engineering’ to create a uniform point of view for SoS high-level security properties. This work is conducted in the Artemis project ARROWHEAD and contributed to the ARROWHEAD framework [1]. Second, we investigate how to represent aggregated information in our assurance approaches [2], in the FP7 project SECCRIT (Secure Cloud Computing for Critical Infrastructure IT).

A first publication [3] related to safety and security co-engineering presents an evaluation of the methods in isolation. For succeeding activities the security analysis an approach based on the ISO 27005 and ETSI TS 102 165-1 standards is used in recent work in ARROWHEAD. For the safety and reliability analysis the IEC 60812 standard is used. Both include an identification of unsatisfactory situations (threats and failure modes) and a method for identifying those with the highest risks. The system is modelled using a dataflow diagram for identifying threats and to motivate decisions when extracting failure modes from an existing catalogue. We have performed an applicability analysis on the resulting threats and failure modes to filter out the relevant ones. In the end the risks of the remaining threats and failure modes were evaluated in detail. The elicitation of threats was supported by a series of workshops and interviews. Results have been applied to current design of one of the project’s pilots. So far we have conducted safety and security analysis individually, and will extend the range of methods. The next step will involve modelling the process and investigating how to describe results to conduct a combined analysis to develop safety and security co-engineering, the fundamentals of which will be contributed to the ARROWHEAD framework.

We have systematically modelled security metrics for Cloud systems to contribute to our assurance model (as introduced in [2]). I.e. ISO27002, defines ‘high-level’ security metrics such as strong passwords. This can be measured by checking if corresponding tools (e.g. PAM (see Link) are available in the constituent components. A catalogue of high level security metrics is being developed and corresponding tool-support will be provided.

Promising initial results have already been published, and form a basis of our research agenda. They will be extended in future projects (e.g. H2020 CREDENTIAL).

Link:
http://www.linux-pam.org/Linux-PAM-html/Linux-PAM_MWG.html
http://www.arrowhead.eu/
http://www.seccrit.eu

References:

[1] S. Plosz, M. Tauber, P. Varga: “Information Assurance System in the Arrowhead Project”, ERCIM News No. 97, pp 29, April 2014.
[2] A. Hudic et al.: “Multi-layer and multi-tenant cloud assurance evaluation methodology, in International Conference on Cloud Computing Technology and Science (CloudCom-2014), 2014.
[3] S. Plósz et al.: “Security Vulnerabilities And Risks In Industrial Usage Of Wireless Communica-tion”, ETFA 2014, September 2014.

Please contact:

Markus Tauber, AIT, Austrian Institute of Technology, Austria
E-mail: markus.tauber@ait.ac.at

Communication and Compatibility in Systems of Systems: Correctness-by-Construction

by Maurice ter Beek, Josep Carmona and Jetty Kleijn

Society is still trying to catch up with technology in the wake of the digital revolution of the last twenty years. Current systems need to be both heterogeneous and able to deal with enormous volumes of data coming from uncertain environments; consequently it is essential to be able to automatically assess the correctness of interactions. To guarantee that a system of systems, comprising a conglomerate of cooperating reactive components, can be trusted, and that the system as a whole behaves as intended, requires a thorough understanding of its communication behaviour. Once local interactions are identified, abstractions can support the identification of incompatibility of systems that should cooperate within a larger system.

In an increasingly smart, connected world in which digital communications outnumber all other forms of communication, it is important to understand the complex underlying interconnections in the numerous systems of systems that govern our daily life. This requires a deep understanding of all kinds of different communication and collaboration strategies (e.g. client-server, peer-to-peer and master-slave) used in embedded or multi-component systems and the risk of failures they entail (e.g. message loss and deadlocks can have severe repercussions on reliability, safety and security).

A project involving ISTI-CNR and Leiden University (the Netherlands) considers fundamental notions paramount for the development of correct-by-construction multi-component systems. Basic building blocks are reactive components that interact with each other via shared (external) actions; internal actions are never shared. External actions can be input or output to the components to which they belong. Components can be added in different phases of construction allowing for hierarchically composed systems of systems. To establish that components within a system or a system and its environment always interact correctly, a concept of compatibility is needed. Compatibility represents an aspect of successful communication behaviour, a necessary ingredient for the correctness of a distributed system. Compatibility failures detected in a system model may reveal important problems in the design of one or more of its components that must be repaired before implementation.

In [1] a definition is given for compatibility of two components that should engage in a dialogue free from message loss and deadlocks. Message loss occurs when one component sends a message that cannot be received as input by another component, whereas deadlock occurs when a component is indefinitely waiting for a message that never arrives. The aim of the ideas developed in [1] is to provide a formal framework for the synthesis of asynchronous circuits and embedded systems. There the approach is restricted to two components and a closed environment, i.e. all input (output) actions of one component are output (input) actions of the other component.

In [2] this approach is generalized to distributed systems which consist of several components, and within which communication and interaction may take place between more than two components at the same time (e.g. broadcasting). These multi-component systems are represented by team automata [3], originally introduced to model groupware systems. Team automata represent a useful model to specify intended behaviour and have been shown to form a suitable formal framework for lifting the concept of compatibility to a multi-component setting. They resemble the well-known I/O automata in their distinction between input (passive), output (active) and internal (private) actions, but an important difference is that team automata impose fewer a priori restrictions on the role of the actions and the interactions between the components [3]. In [2] emphasis is on team automata with interactions based on mandatory synchronized execution of common actions.

Together with the Universitat Politècnica de Catalunya (Barcelona, Spain) we plan to continue the approach of [2] by investigating other composition strategies and, in particular, focusing on how to handle compositions based on master-slave collaborations. In such collaborations, input (the slave) is driven by output (the master) under different assumptions ranging from slaves that cannot proceed on their own to masters that should always be followed by slaves. Thus we address questions such as “how is compatibility affected when slaves are added?” and “in what way does compatibility depend on the collaboration among slaves?” Practical solutions to these answers may have strong impacts in various fields, such as services computing and security.

Composition and modularity are common in modern system design. So compatibility checks considering varying strategies significantly aid the development of correct-by-construction multi-component systems. Hence the ideas in this project should serve the development of techniques supporting the design, analysis and verification of systems of systems.

References:

[1] J. Carmona and J. Cortadella: “Input/Output Compatibility of Reactive Systems”, Formal Methods in Computer-Aided Design, LNCS 2517 (2002) 360-377

[2] J. Carmona and J. Kleijn: “Compatibility in a multi-component environment”, Theoretical Computer Science 484 (2013) 1-15

[3] M.H. ter Beek and J. Kleijn: “Modularity for Teams of I/O Automata”, Information Processing Letters 95, 5 (2005) 487-495

Please contact:
Maurice ter Beek
ISTI-CNR, Italy
E-mail: maurice.terbeek@isti.cnr.it

Safety Analysis for Systems-of-Systems

by Jakob Axelsson

The introduction of systems-of-systems (SoS) necessitates the revision of common practices for safety analysis. In the case of vehicle platooning, for instance, this means that an analysis has to be carried out at the platoon level to identify principles for the safety of the SoS, and these principles then have to be translated to safety goals and requirements on the individual trucks.

The term systems-of-systems (SoS) started to become relevant some 20 years ago, and accelerated as a research area around 10 years ago. Although some people tend to take SoS as a synonym for large and complex systems, the research community has arrived at a fairly precise characterization of the term: in an SoS, the elements, or constituent systems, exhibit an operational and managerial independence, meaning that they can operate outside the SoS context, and have different owners. They choose to collaborate in order to achieve a common goal, manifested as an emergent property of the SoS, i.e. a property that does not exist in any of its parts in isolation. A recent literature review [1] shows that the field, so far, has been dominated by US researchers focusing on military and space applications. Key topics include: architecture, communications, interoperability, modelling and simulation, and also a number of properties where dependability attributes, such as safety, play an important role.

From its origins in the government driven sectors, SoS are now spreading