Ph.D. in Information Technology: Final Dissertations

DEIB Conference Room

February 7th, 2017

10.00 am

First Ph.D. presentation and discussion

Naser DERAKHSHAN – XXIX Cycle

"A Middleware for Peer-to-Peer Interaction of Smart Things"

Advisor: Prof. **Luciano Baresi**

Abstract:

In these years, we are assisting to the rapid development of small, low-power, and low-cost wireless computation/communication devices, which have served as enablers for the so-called "Internet of Things" (IoT). Devices that can connect to the Internet can exploit cloud-based services to enable interaction in an IoT scenario. However, an Internet connection might not always be available, or it might be too expensive to use. It is clear that within the Internet of Things we cannot impose the requirement of having an always-on functional Internet connection. Instead, we need to shift our focus to a new wave of interaction called "proximity-based interactions". In this kind of interactions, applications operate in an infrastructure-less scenario, possibly interacting in a peer-to-peer (P2P) manner. Among P2P communication protocols for mobile devices, Wi-Fi Direct has recently gained attention. Although many works have already tried to exploit Wi-Fi Direct in social interactions among proximal smart devices, there is no work to the best of our knowledge that attempts to exploit Wi-Fi Direct in large-scale dynamic application domains.

This thesis bridges this gap by proposing a middleware infrastructure, called "MAGNET", a novel middleware infrastructure that exploits Wi-Fi Direct to provide a reliable and stable communication means for large numbers of mobile devices. To evaluate the effectiveness of MAGNET, a Wi-Fi Direct simulator has also been developed, called "WiDiSi". WiDiSi's main goal is to allow Android Wi-Fi direct applications to be easily tested in large-scale dynamic scenarios. The proposed solution has been tested on real eighteen Android devices and on the simulation environment. The evaluation results illustrate the effectiveness of such a solution in providing a stable communication between many mobile devices using Wi-Fi Direct.

Second Ph.D. presentation and discussion

Luca FLORIO – XXIX Cycle

“Design and Management of Distributed Self-Adaptive Systems”

Advisor: Prof. **Elisabetta Di Nitto**

Abstract:

Self-adaptation is the capability of a system to adapt in an autonomous way to every change in the scenario where it operates. This capability is fundamental in distributed systems, which plays a central role in the current software development landscape, and several algorithms for self-adaptation of these systems have been proposed in literature. However, stable platforms and comprehensive software engineering approaches for the application of such algorithms to a concrete context are still to come. This thesis addresses these challenges, providing methods and tools to design and manage distributed systems able to autonomously adapt to changes and operate in a concrete and dynamic context. In order to validate our research, we designed two decentralized self-adaptive systems that have been applied to the domain of cloud-based applications and microservices. The evaluation of our prototypes has been done deploying a concrete case study in a cloud computing infrastructure, and the results validate our approach for the application of self-adaptation to distributed systems.

Third Ph.D. presentation and discussion

Alessandro Maria RIZZI – XXIX Cycle

“A Syntactic - Semantic Approach for Incremental Program Verification of Matching Logic Properties”

Advisor: Prof. **Carlo Ghezzi**

Abstract:

SOFTWARE is not built monolithically; instead, its construction requires a sequence of small steps. In fact, software is continuously subjected to changes; either in the specification and requirements, due to changes in the environment or in user needs; either in the implementation, in order to fix a software fault or to implement a missing functionality. In both cases, change is a distinctive feature of software systems, affecting almost every phase of its life-cycle. On the other hand, formal software verification techniques are becoming more and more effective in helping the developer of software systems. However, formal verification technology is still not widely adopted for ensuring software correctness. One of the main reason for this is the high demand of computation resource which affects every formal verification techniques due to the high computational complexity of the problems involved. In our opinion, it is possible to exploits the incremental

nature of software systems to develop more efficient verification techniques. Many of the existing formal verification approaches do not exploit the fact of having a sequence of different software versions which change relatively little one to another; instead they reapply the verification process from scratch to each version. We address this problem by developing formal verification techniques capable of incrementality: techniques which are able to reuse the intermediate results obtained with the verification of an early version of a software system to the verification of the new versions. Our goal is to provide formal verification as a feasible and practical way of assessing quality in software products. Nowadays, the most applied verifi- I cation technique is testing which has gained
popularity with the introduction of Agile methodologies. It is applied to evolving software systems in the form of continuous testing: software quality is assessed by automatically applying a set of tests to each version of a product. Our ultimate goal is to provide an incremental formal verification approach which can complement testing, as a practical and reliable verification technique still providing strong formal guarantees of correctness. We target incremental verification techniques based on the syntactical structure of the software artifact being verified. The formal language which defines the software artifact can discover the changes occurred at a syntactical level: Software artifacts are expressed by operator-precedence grammars, which, thanks to properties such as locality, provide ways to efficiently reparse only the parts affected by the change.
This incrementality obtained at the syntactic level is then translated to semantics by means of the attribute grammar formalism. In particular, by using synthesized-only attribute grammars, the semantic effect of a change is limited to a portion of the abstract syntax tree defined by the program. This idea has been developed in the SiDECAR framework, a system for defining a incremental verification techniques on a given programming language. In this work, this general technique is applied to matching logic properties verification. Matching logic is a Hoare-like verification system rooted on a formal definition of programming language semantics. We have analyzed the case of KernelC language. KernelC is a subset of the C language having interesting features as heap, loops, function calls and recursion. In this work, we have developed a matching logic verification framework expressed as an evaluation of semantic attributes. Upon this approach we have built our incremental verification technique. This approach has been implemented using the framework SiDECAR, for the incremental matching logic verification of KernelC programs. We experimentally evaluated our incremental verification approach over a large suite of programs, containing many versions of each program; along with the state-of-the-art, non-incremental tool MAT C HC for the verification of KernelC programs annotated with matching logic properties. In particular we first evaluate the non-incremental performance of our tool, comparing it with MAT C HC over the examples MAT C HC is provided with. Then, we evaluate the approach with incrementality over programs either constructed applying random mutation to MAT C HC examples, or considering well known benchmarks, such as the Siemens test suite and classic literature data structures, as red-black trees. The obtained results show that our incremental approach can

effectively speedup software verification process, in the case of matching logic verification of KernelC programs.

Fourth Ph.D. presentation and discussion

Christos TSIGKANOS - XXIX Cycle

"Modelling and Verification of Evolving Cyber-Physical Spaces"

Advisor: Prof. **Carlo Ghezzi**

Abstract:

Computing and communication capabilities are increasingly being embedded into physical spaces thus blurring the boundary between computational and physical worlds; typically, this is the case in modern cyber-physical systems, like smart buildings or smart cities. Conceptually, such composite environments, hereafter called cyber-physical spaces (CPSp), can be abstracted into a topological model where computational and physical entities are connected in a discrete, graph-like structure. Like any other software-intensive system, a CPSp is highly dynamic and typically undergoes continuous change - it is evolving. This brings a manifold of challenges as dynamics may affect safety, security, or reliability requirements of the overall system. Formally modelling space and its dynamics as well as supporting reasoning about various properties of evolving space, is a crucial prerequisite for engineering dependable evolving CPSp, e.g. to assure requirements satisfaction or to trigger correct adaptation. We propose a methodology and technical framework which support modelling of evolving cyber-physical spaces and reasoning about their spatio-temporal properties.

We utilise bigraphs as a formalism for modelling CPSp as well as primitives of change, giving rise to a reactive system consisting of rewriting rules with both local and global application conditions. Formal reasoning facilities feature adopting logic-based specification of properties and according model checking procedures, in both spatial and temporal fragments. Utilising these foundations, we consider systematically engineering systems and support both verification of requirements in early stages of design and devising adaptive security behaviours at run time.

Fifth Ph.D. presentation and discussion

Eric UMUHOZA – XXIX Cycle

"Domain-specific Modeling and Code Generation for Cross-platform Mobile and IoT-Based Applications"

Advisor: Prof. **Marco Brambilla**

Abstract:

Nowadays, mobile devices constitute the most common computing device. This new computing model has brought intense competition among hardware and software providers who are continuously introducing increasingly powerful mobile devices and innovative OSs into the market. In consequence, cross-platform and multi-device development has become a priority for software companies that want to reach the widest possible audience. However, developing an application for several platforms implies high costs and technical complexity. Currently, there are several frameworks implementing different methodologies for cross platform application development.

Nevertheless, these approaches still require manual programming, which yields to high risks of errors, inconsistencies, and inefficiencies. On the other side, with the advent of the Internet of Things era, homes, cities and almost everything is becoming smart. This technology is extending very rapidly, but it still needs to solve many problems that arise when a technology requires being available at any time, in any place, for everyone and for any device.

So far the development of front-end and user interfaces for IoT systems has not played a relevant role in research. On the contrary, user interfaces must be a key part in the IoT ecosystem because they can play a key role in acceptance of solutions by final adopters. This research proposes to face the challenges of mobile and IoT-based applications development by exploiting abstraction, modeling and code generation in the spirit of model-driven development engineering. The main contributions include modeling languages and design methodology for mobile and IoT-based applications; code generators for implementation phase; and a model-driven framework for user behavior analysis. In parallel to the main topic of this thesis, I carried out different experiments in order to understand and propose solutions to some supposed barriers in the adoption of model-driven development approaches. In this context, I tackled the issues of modeling effort and modeling languages usability. Regarding the modeling effort in the process of modeling software systems showed that more than 60% of effort is spent in designing. Thus, the fault of supposedly unproductive processes should not be blamed on modeling, but to the (anyhow necessary) time devoted to thinking about the problem and identifying the solution. Furthermore, I conducted a research to understand how the designed languages are actually used and how they fit the users' need. This investigation showed that available languages are either too complex (e.g.: BPMN) with respect to user need or do not exactly fit the domain (e.g.: UML) and this happens mainly because of the lack of right involvement of end-users in the language development process. I proposed a user-centered approach allowing the adaptation of existing modeling languages to the user needs through a language simplification process.

PhD Committee:
Prof. **Marco Brambilla**, DEIB – Politecnico di Milano
Prof. **Paolo Bellavista**, Università di Bologna
Prof. **Antonio Filieri**, Imperial College London