- Automated model-based Software Performance Engineering through system trace logs

Performance models of software designs can allow the early discovery of problems such as over utilized servers or slow response times. However, the development cost of these models is high, due to the need of both manual effort and performance knowledge to develop them; fact that can prevent the usage of performance models in practice. To facilitate this development, semi-automatic tools have been developed in the academic field, allowing the derivation and analysis of performance models without requiring extensive knowledge of analysis methodologies to the application designer.

Goal of this project is the realization of L(og)4Klapersuite as an extension of Klapersuite, one of the existing tools. This enhanced version would exploit the information available in existing trace logs to complete the definition, in an automatic way, of the software performance models. Trustworthiness analysis of the results will be included through a rigorous definition of mathematical methods able to deal with uncertainties present in the trace logs. The human intervention in such a way is reduced to the effort of the Software Architect who should provide the software architecture models. Then, by exploiting the Klapersuite features, it will be possible to have in a single step a KLAPER model and finally a performance model ready to be evaluated.

- Utility-Aware Fully Decentralized Service Assembly

Modern applications, like those envisioned in pervasive computing scenarios, are increasingly reliant on systems built from multiple distributed components, that must be suitably composed to meet some specified functional and non functional requirements. A key challenge for the software engineering community is how to efficiently and effectively manage such complex systems. The use of autonomic self-management capabilities has been suggested as a possible way to address this challenge. To cope with the scalability and robustness issues of large distributed systems, self-management capabilities should ideally be architectured in a decentralized way, where the overall system behavior emerges from local decisions and interactions.

Within this context, fully decentralized solution to the adaptive self-assembly problem of distributed services are needed. Goal of this project is to devise solutions, based on gossip protocols and/or bio-inspired techniques, able to build and maintain an assembly of services that, besides functional requirements, is able to fulfill also global quality of service (QoS) requirements and structural properties.

- Analysis of Bursty Workload-aware Self-adaptive Systems

Software is often embedded in dynamic contexts where it is subjected to high variable, non-stable, and usually bursty workloads. A key requirement for a software system is to be able to self-react to workload changes by adapting its behavior dynamically, to ensure both the correct functionalities and the required performance.

Research on fitting variable workload traces into formal models has been carried out using Markovian Modulated Poisson Processes (MMPP). These works concentrate on modeling stable workload states, but accurate modeling of transient times still deserves attention since they are critical moments for the self-adaptation.

Goal of this project is to build on research in the area of MMPP trace fitting and investigate other possible solutions that also accounts for transient times.

- On the Relationships between QoS and Software Adaptability at the Architectural Level

Modern software operates in highly dynamic and often unpredictable environments that can
degrade its quality of service. Therefore, it is increasingly important having systems able to adapt their behavior. However, the achievement of software adaptability can influence other software quality attributes, such as availability, performance or cost. This paper proposes an approach for analyzing tradeoffs between the system adaptability and its quality of service. The proposed approach is based on a set of metrics that allow the system adaptability evaluation.

The approach can help software architects to guide decisions on system adaptation for fulfilling system quality requirements.

Goal of this project is the development of a framework that, using the proposed metrics, guides the software architects in the selection of the most suitable software architecture.

- Uncertainties in the modeling of self-adaptive systems: performance and availability examples

The complexity of modern software systems has grown enormously in the past years with users always demanding for new features and better quality of service. The satisfaction of non-functional requirements like performance and availability is of paramount importance if a product hopes to be considered in the marketplace. Model-based evaluation techniques at software design-time have been proposed to ensure the delivering of software that meets its non-functional requirements. However, since a large part of modern software is embedded in dynamic execution contexts where requirements, environment assumptions, and usage profiles continuously change, this quality assessment at design time becomes more difficult.

Goal of this project is to 1) identify the existence of uncertainties in the knowledge of the system behavior, 2) analyze them to unveil which type of uncertainty is and 3) propose techniques to manage the system considering the awareness that some information is uncertain for the satisfaction of availability and performance requirements.