behavioural properties, resource-management policies and self-adaptation strategies can be clearly specify and implement component coordination on a high-level of abstraction, in such a manner that self-adaptation mechanisms along with appropriate specification languages, allowing designers to design approach—to be implemented in JavaBIP—which would provide resource management and one hand, and adaptation policies, on the hand. These models will form the foundation of a rigorous Theoretical work within the project will aim to develop expressive formal models for the specification with information about resource availability to optimise overall system performance. requirements to the coordinating engines, which in turn must be able to combine such requirements framework. To enable efficient coordination, components must advertise their resource represent resource availability and dependencies must be explicitly provided in the design variable. In particular, this is due to interferences among applications sharing common resources and management and self-adaptation. Indeed, the environment of modern systems is inherently highly fluctuation of resource availability become the norm rather than an exception. Instead of waiting for the resources to be available, applications adapt their behaviour to the changes in the environment. Mechanisms representing resource availability and dependencies must be explicitly provided in the design framework. To enable efficient coordination, components must advertise their resource requirements to the coordinating engines, which in turn must be able to combine such requirements with information about resource availability to optimise overall system performance.

Theoretical work within the project will aim to develop expressive formal models for the specification and analysis of platform capacities and application requirements for various kinds of resources, on one hand, and adaptation policies, on the hand. These models will form the foundation of a rigorous design approach—to be implemented in JavaBIP—which would provide resource management and self-adaptation mechanisms along with appropriate specification languages, allowing designers to specify and implement component coordination on a high-level of abstraction, in such a manner that behavioural properties, resource-management policies and self-adaptation strategies can be clearly
stated, combined and enforced.

References


Main activities

Main activities:

- Scientific research (definition of models, algorithms etc.; proofs)
- Implementation of prototype tools for evaluation of the proposed techniques
- Written presentation of the obtained results through papers and reports
- Oral presentation of the obtained results at scientific conferences
- Participation in the supervision of students at all levels

Additional activities:

- Strengthening of one's scientific network and definition of a career strategy
- Participation in the development of a user community
- Participation in other activities to promote the team's research to broader audiences

Skills

- Formal methods (in particular semantic models, e.g. finite automata, Labelled Transition Systems and Petri Nets; behavioural equivalences, e.g. trace equivalence and bisimilarity)
- Verification (in particular temporal logics, e.g. LTL and CTL; tools, e.g. nuXmv, mCRL2)
- Knowledge of coordination languages, such as BIP, is a plus
- Proven experience in preparation of scientific documents (including mastery of LaTeX)
- Proven experience in software development (Java, Python)
- Excellent communication skills

Benefits package

- Subsidised catering service
- Partially-reimbursed public transport

Remuneration

Around 31 000 € yearly brutto.