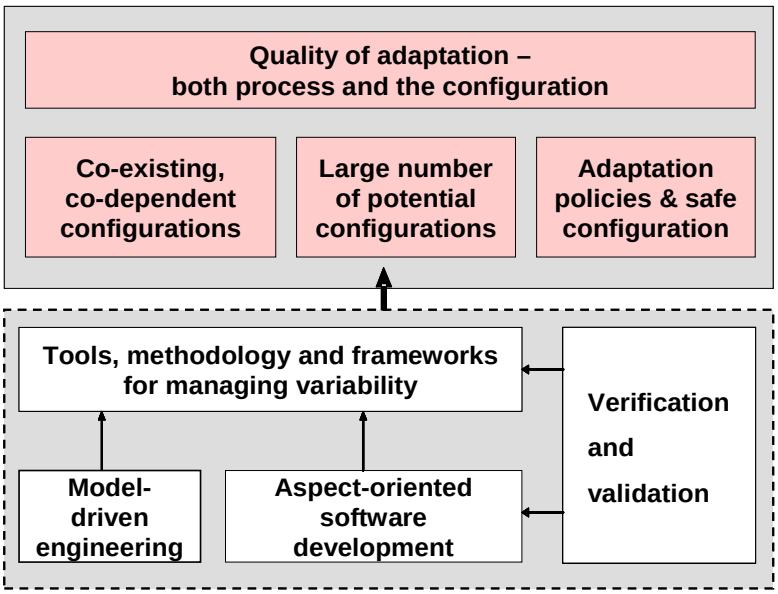
Dynamic Variability in complex, Adaptive systems

Objectives

DiVA research objectives

1. To provide novel build time and runtime management of adaptive system (re)configuration of co-existing, co-dependent configurations that can span across several administrative boundaries in a distributed, heterogeneous environment.
2. To provide efficient handling of the number of potential configurations, that may grow exponentially with each new variability dimension,
3. To increase quality and productivity of adaptive system development and help the designers to model, control and validate adaptation policies as well as the trajectory going from one safe configuration to another.



Approaches to the objectives

Run-time management: The first objective will be accomplished by leveraging the synergy between model-driven and aspect-oriented techniques. There will be a focus on separating the application-specific functionality from the adaptation concerns. Aspect-oriented techniques will be utilised to analyse and reconfigure crosscutting features dynamically. Model driven techniques will be used to raise the level of abstraction and to provide models at runtime. In DiVA, runtime models will expose access points for dynamic manipulation and adaptation. The current AOSD (Aspect Oriented Software Design) and MDE (Model Driven Engineering) techniques will be extended with support for dynamic variability and for reasoning about problems of co-dependent co-existing system configurations.

Managing the exponential growth: The second objective will be addressed by efficient analysis, design and runtime representations of potential configurations using aspect-oriented modelling techniques and by using model composition as a variability mechanism. Furthermore, DiVA will provide analysis techniques to reduce the number of configurations to the most pertinent ones.

Increased system development productivity: The third objective will be accomplished by providing an integrated toolset, a methodology and frameworks. The variability and validation analysis will be supported by a methodology for development of variability requirements, design and code artefacts, and analysis of their configurations and co-dependencies, and selection of preferred alternatives, thus, progressively reducing the alternatives' space. Moreover, DiVA intends to provide techniques for verifying and validating the adaptation both in terms of realising the goals of the adaptation and its trajectory.

Pilot case studies: The fourth objective will be accomplished by implementing case studies from two different domains: crisis management and Customer Relationship Management (CRM), and by disseminating the results through various dissemination channels including publications at international conferences and workshops, contribution of technologies to the open source communities.

SEVENTH FRAMEWORK PROGRAMME

SINTEF

INRIA

THALES

CAS SOFTWARE

LANCASTER UNIVERSITY

pure-systems

Challenges

The dynamicity of adaptive systems makes existing software engineering techniques difficult to apply. In a traditional software development cycle all the behaviours of the system must be captured at design time while for an adaptive system the behaviour of the system has to be dynamically adapted to an evolving environment. There is a need to manage the variability all along the life-cycle of an adaptive system: from the requirements elicitation and analysis to design, test and maintenance. The variability especially needs to be carefully modelled and validated. Trustworthiness is a major concern for these kinds of systems. Three important issues in the modelling of an adaptive system are:

- **Specifying and modelling variations in adaptive systems:** Modelling software variations is a challenge that has already been studied in the context of software product lines. During requirements analysis and design a set of variation points is identified to capture the required variations. Then, sets of variants are defined to implement the strategies associated with each variation point. A similar process could be applied to adaptive system modelling. However, the way variation is used in the context of product lines and for an adaptive system is different. For product lines a particular set of variants is chosen at compile-time (or at deployment-time) whereas for an adaptive system variations have to be available and managed also at runtime.
- **Managing the variant explosion in adaptive systems:** Another challenging issue is the explosion of the number of potential variants with each new variability dimension. One approach is to pre-calculate the properties of each potential variant and thereby be ready to select the most appropriate one when the context changes occur. Another approach is to determine only the variability dimensions and their properties and dependencies, and do the calculation of properties at deployment-time. Obviously such calculations take some time, so trade-offs have to be made between performance and the possibility to select from all possible variants.
- **Dealing with the co-existing, co-dependent configurations:** An adaptive system in a heterogeneous environment involves explicit or implicit co-existing, co-dependent configurations that often include pre-existing or even third party software components. The running system is nevertheless supposed to be able to expose its ability to be dynamically adapted to changing conditions in the real world. Techniques for addressing dynamic variability not only need to incorporate support for modelling such evolving co-existing, co-dependent configurations but also the ability to choose optimal adaptation paths in the presence of such configurations. Furthermore, reasoning about emergent behaviour resulting from the physical and social context in which multiple, co-dependent configurations must be deployed.

Meeting the challenges

- The main idea behind the DiVA project is to model variability dimensions as aspects and do dynamic aspect weaving as model transformations when an adaptation need to occur. DiVA will concentrate on the adaptation part of a system, the system itself being from the point of view of DiVA a kind of grey-box component consisting of services that may have well defined adaptation interfaces.
- Some examples of variability dimensions are security, communication, availability and performance. Examples of security features (modelled as aspects) are authorization and authentication; the communication dimension can provide several options for communication such as WiFi, Ethernet, and Modem; replication can be provided as a feature supporting availability and features for compression and decompression can be provided to increase performance in some contexts.

DiVA will provide a new tool-supported methodology with an integrated framework for managing dynamic variability in adaptive systems and combines aspect-oriented and model-driven techniques.

DiVA is supported by the European Commission in the 7th Framework Programme, Project Reference: 215412.