Families of Dependable Systems: A Model Checking Approach

Stefania Gnesi
ISTI-CNR, Pisa, Italy

Software Product Line Engineering (SPLE) is a paradigm for developing a diversity of software products and software-intensive systems based on the underlying architecture of an organisation's product platform. In the context of Software Product Lines (SPLs) the introduction of variability in the software development cycle has been extensively studied [5, 7]. At all abstraction levels, a product line description is composed of a constant part and a variable part. Variability among products is made explicit by variation points, i.e., places in design artifacts where a specific decision is reduced to several features but the feature to be chosen for a particular product is left open (like optional, mandatory, or alternative features). Variety from a single product platform is achieved by identifying such variability points. Variability management is the key aspect differentiating SPLE from conventional software engineering.

Modelling variability in product families has been studied extensively in the literature on SPLs, especially that concerning feature modeling [6]. Formal methods have been developed to show that a product belongs to a family or to derive instead a product from a family. Deontic-style logics [1, 8] have become popular to formalize descriptional and behavioural aspects of computer systems, mainly because they provide a natural way to formalise concepts like violation, obligation, permission and prohibition. Intuitively, these concepts permit one to distinguish correct (normative) states and actions from non-compliant ones. Hence, deontic logic is a natural candidate for expressing the conformance of members of a family of products with respect to variabilities.

A number of models, logics and associated tools for the qualitative analysis of variability aspects and their use to deal with adaptability and evolvability of systems have recently been proposed. In these lectures, we will focus on the approach presented in [2, 3, 4], where the introduction of the action-based branching-time temporal logic MHML allows expressing constraints over the products of a family as well as constraints over their behaviour in a single logical framework. Based on model-checking techniques for MHML, a modelling and verification framework will be presented that can automatically generate all the family's valid products, visualise the family/products behaviour and efficiently model check properties expressed in MHML over products and families alike.

The use of the above methods, techniques and tools will be applied to a scenario derived from a family of dependable systems.

References

[1] L. Åqist. *Deontic Logic.* In: D. Gabbay, F. Guenthner (eds.), Handbook of Philosophical Logic, 2nd edition, Vol. 8, pp. 147-264; Kluwer; 2002.

[2] P. Asirelli, P., M.H. ter Beek, A. Fantechi, S. Gnesi. *A Logical Framework to Deal with Variability.* In: D. Méry, S. Merz (eds.), IFM 2010, LNCS 6396, pp. 43-58; Springer; 2010.

[3] P. Asirelli, M.H. ter Beek, A. Fantechi, S. Gnesi; *A Verification Environment for Families of Services*. In: R. Bruni, J. Dingel (eds.), FMOODS/FORTE 2011, LNCS 6722, pp. 44-58; Springer; 2011.

[4] P. Asirelli, P., M.H. ter Beek, S. Gnesi, A. Fantechi. *Formal Description of Variability in Product Families*. In: SPLC 2011; pp. 130-139; IEEE; 2011.

[5] P.C. Clements, L. Northrop. *Software Product Lines – Practices and Patterns*. Addison-Wesley; 2002.

[6] K. Kang, S. Choen, J. Hess, W. Novak, S. Peterson. *Feature Oriented Domain Analysis (FODA)*; Feasibility Study; Technical Report SEI-90-TR-21; Carnegie Mellon University; 1990.

[7] K. Pohl, G. Böckle, F. van der Linden. *Software Product Line Engineering – Foundations, Principles, and Techniques*. Springer; 2005.

[8] J.-J.Ch. Meyer, R.J. Wieringa (eds.). *Deontic Logic in Computer Science – Normative System Specification*. John Wiley & Sons; 1994.