WP7: Incremental design and verification

Initial partners: UCL, ULB, FUNDP
Coordinator: Pierre-Yves Schobbens

MoVES annual meeting
Basic idea

★ Systems evolve mainly by adding features
★ Changes are often incremental
★ Radical restructuring is rare, and often refactoring:
  no added (visible) functionality, clear mapping
★ Thus we hope to reuse designs and proofs
★ Might require to structure them differently:
  « evolution-ready »
Main topics this year

★ Incremental specification and verification of software product lines
  (FUNDP, UCL, ULg and ULB)
★ Developing feature diagrams verification tools
  (FUNDP)
★ Incremental synthesis for temporal logic
  (ULB, FUNDP)
MoVES-Verif@ASE 2010, September 21st

★ Realizability of real-time logics: decidability and implementation
  L. Doyen, B. Di Giampaolo, G. Geeraerts, J.-F. Raskin, J. Reichert and N. Sznajder

★ Decidable Distributed Event Clock Automata,
  J. J. Ortiz and P.-Y. Schobbens

★ Invited Speaker: Rupak Majumdar, UCLA

★ Implementing High-Level Languages for Software Product Line Model Checking,
  A. Classen, P. Heymans, A. Legay and P.-Y. Schobbens

★ Combining Partial Order Reduction and Symbolic Model Checking to verify LTL properties,
  J. vander Meulen

★ Parameterized Verification of Ad Hoc Networks,
  A. Sangnier

★ Symbolic Analysis of Concurrent Programs with Polymorphism
  N. S. Rungta

★ Modular Lightweight Semantics
  K. Madlener, S. Smetsers and M. van Eekelen

★ Evolutions of Test Systems,
  P. Y. H. Wong and N. Diakov
Newsletter n°2

★ 9 articles
★ 3 WP7 themes

MoVES Newsletter
Work Package 7 - Incremental design and verification
University in Focus: University of Namur

Editorial

Dear Reader,

This is the second issue of the MoVES newsletter. MoVES stands for ‘Modelling, Verification and Evolution of Software’ and addresses these fundamental issues in software engineering. The project is sponsored by the Belgian government (belspo IAP programme). Each issue of the newsletter presents a partner and a work package.

This issue presents results and ongoing research of Work Package 7 - incremental design and verification. The idea of WP7 is to bridge the main topics of the project. Evolution poses a challenge for both modelling and verification: we want to reuse the modelling and verification effort as much as possible when evolving a software system. This requires innovative structuring of the software and the verification process. This issue also presents the PReCISE Research Centre of the University of Namur (FUNDP), and shows the variety of research pursued there.

Enjoy reading!

Pierre-Yves Schobbens, Nicolas Genon and Andreas Classen

Upcoming Events & Recent Joint Publications

- Moon Boat, 21 September.
- ACM/IEEE 13th International Conference on Model Driven Engineering Languages and Systems, MODELS 2010, 3-4 October, Oslo, Norway.
- ME 2010 International Workshop on Models and Evolution, 3 October.
  - A Lightweight Sanity Check for Implementer Architectures. A. van Deursen and E. Bouwers.
- Submission deadlines:
  - October 31st 2010 – Special Issue on “Automated Software Evolution” of the Elsevier Journal on Systems and Software.
  - February 24th, 2011 – 16th IEEE International Requirements Engineering Conference, RE 2011, August 29th - September 2nd 2011, Trento Italy. The theme will be “Requirements in Motion”.

WP7 – GM 7 Feb 2011
Incremental verification of software families (FUNDP-UCL-Ulg-ULB)

★ Ideas:

‣ Develop verification techniques for model-checking adapted to software families
‣ Based on a behaviour model tagged by features
‣ Checking the whole family in one shot
‣ Specialised data structures
‣ Both symbolic and enumerative techniques
‣ High-level language
Incremental verification of software families (FUNDP-UCL-ULg-ULB)

★ Presented in:
  ‣ Classen, A.; Heymans, P.; Schobbens, P-Y. and Legay, A. Symbolic Model Checking of Software Product Lines. Accepted for publication at the 33rd International Conference on Software Engineering, ICSE 2011.
Automated verification of feature diagrams (FUNDP)

★Use feature diagrams verification tools (based on SAT or SMT solvers) when generating new programme variants
  ‣ Model-check programme variants (feature combinations)
  ‣ Compute various indicators on legal programme variants (number of such programmes, changes that are always necessary, co-occurring changes, etc.)
  ‣ Study of configuration workflows
  ‣ Computation of feature diagram views
Incremental synthesis for temporal logic

- Synthesis of strategies (= programs) in a game with an objective in temporal logic
- The controller must guarantee the objective even in face of arbitrary environment perturbations
- Doubly-exponential but works on examples
- Extended to real-time logics
- New implementation in project
Future plans

★ Pursue and extend current topics
★ Stronger links with Aspects, compositionality
★ Extend with real-time
★ Joint events with WP5 and WP6
★ More common theses