Making Software Fulfill Users Goals: From Goals to Code

Oscar Pastor

Centro de I+D en Métodos de Producción de Software (PROS)
Universitat Politècnica de València
València, Spain
opastor@dsic.upv.es

Abstract. Goal-oriented Requirements Engineering (GORE) has conformed in the recent years a sound research working domain, where the *i** ([1]) approach is being playing a leading role. The idea of starting any software process with a well-founded goal modeling process is widely accepted, as it is essential for the quality of a software product to know and to trace effectively why software services exist, and what goal(s) justify and explain their existence.

With all this theoretical and practical background, it is the time to focus on an efficient GORE in Practice perspective. This is the main topic of this keynote. By GORE in Practice it is meant to provide a Model-driven development (MDD) approach based on a Model-driven architecture (MDA) and any other accepted Model-driven engineering proposal for defining a software production process- where a strict goal modeling step has to be the very first step of the process. Having *i** as a reliable theoretical reference model, with also successful practical experiences reported ([2]), this first step that corresponds to a Computation Independent Model (CIM) in MDA terms, should not more attention. It is just a matter of using *i**, and extending is use in practice.

But this goal-oriented model needs to be extended to elaborate a requirements model where not only goals, but i) system services that materialize them and ii) data that those services require to occur, are properly specified. This extension can be done using a set of diverse Requirements Engineering (RE) techniques, and in this keynote we will analyze in particular how a BPMN-based notation under a Communication Analysis approach can be used. At the end of this CIM-based step, a full requirements models is available, where user goals and their associated system services are precisely identified.

The next problem to be solved is again in MDA terms- the CIM to PIM (for Platform-Independent Model) transformation. Now, how to face a sound model transformation that takes an extended *i** model as the input model in order to generate its associated executable conceptual schema, becomes the challenge. Under the umbrella of the Conceptual Programming notion ([3], the Conceptual Schema-based Software Development approach ([4]), the Extreme Non-Programming proposals ([5]) and all this type of proposals, we will analyze in the keynote how current tools that provide such a conceptual model compiler (i.e, Integranova

([6]), WebRatio ([7]), ...) can be used to provide the GORE in Practice capabilities introduced before.
Characterizing how to convert an extended goal-model -where the services that materialize any user goal are clearly specified- into an executable conceptual schema makes possible to ensure that any piece of code of the final application is the representation of a corresponding goal. This allows to assess traceability from goals to code, making goals really become the key notion of the proposed Requirements Model-based Software Process.

Keywords: Goal-Oriented Requirements Engineering, istar, Requirements Model-driven based Development, Conceptual Programming, From Goals to Code

References

1. http://www.wikicfp.com/cfp/home
2. E. Yu, D. Amyot, G. Mussbacher, X. Franch and J. Castro, "Practical applications of i in industry: The state of the art," 2013 21st IEEE International Requirements Engineering Conference (RE), Rio de Janeiro, 2013, pp. 366-367.
3. David W. Embley, Stephen W. Liddle, Oscar Pastor: Conceptual-Model Programming: A Manifesto. Handbook of Conceptual Modeling 2011: 3-16.
4. Antoni Oliv: Conceptual Schema-Centric Development: A Grand Challenge for Information Systems Research. CAiSE 2005: 1-15.
5. Morgan, T.: Business Rules and Information Systems Aligning IT with Business Goals. Addison-Wesley, Reading (2002).
6. http://www.integranova.com/
7. https://www.webratio.com/site/content/en/home