TU/e technische universiteit eindhoven

espresso

SoC Software Construction

TABASCO: a Taxonomy-based Domain Engineering Method

Loek Cleophas, Iwan Vosloo & Bruce W. Watson
loek@loekcleophas.com, iwan@global.co.za, bruce@bruce-watson.com

Espresso Research Group / Software Construction Group
http://espresso.cs.up.ac.za   http://www.win.tue.nl/soc

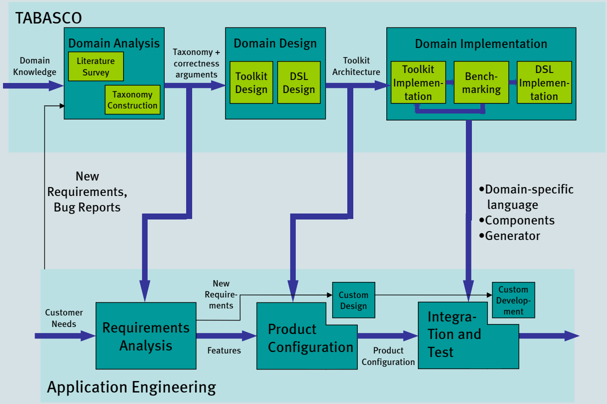
TABASCO is a domain engineering method for a particular kind of domain. It is intended for ordering an algoritmic problem domain and constructing a *Domain Specific Toolkit* (DST) for it.

Domain engineering is often presented as applicable to large business software domains and complex software systems. The application of domain engineering to such fields entails a risk, related to the complexity and third-party development involved. Such domains may not be ready for product family development using domain engineering.

TABASCO is a formal domain engineering approach, restricting a domain to a family of related algorithms or data structures, and assuming no third-party involvement in the process.

The TABASCO process

1. Problem field selection and literature survey to find algorithms solving the problem.
2. Rewriting of algorithms in common style, to make it easier to distill commonalities and variation.
3. Construction of a taxonomy of solutions. Abstract algorithms in the taxonomy are derived from a common starting point by adding correctness-preserving details which indicate algorithm variations. The taxonomy makes it easier to compare algorithms, argue about correctness and efficiency, and may lead to new algorithms.
4. Toolkit development. Domain/toolkit design is not a straightforward task, but the restricted form of our domains and domain models helps. A toolkit design does not directly follow from the taxonomy, but the taxonomy makes algorithm commonalities and variation explicit, showing algorithm groupings as well as how algorithms in a group differ. High-level design choices are thus guided by taxonomy structure, and the choice of language constructs to implement design parts can be based on standard design techniques. The toolkit will be homogeneous and coherent, and give confidence about correctness and efficiency.



Overview of TAxonomy BAsed Software COnstruction

5. Benchmarking can be performed to determine algorithms' practical performance. The results help toolkit users to determine which algorithm satisfies their performance requirements.
6. *Domain Specific Language* (DSL) development. User requirements often focus on performance, taxonomy-based toolkits on correctness and clarity. To overcome this mismatch and simplify component selection, a DSL is developed. This DSL uses concepts the average user is familiar with or interested in to specify component requirements. Such requirements are then mapped to toolkit components. The DSLs developed can range from very basic—merely a few paragraphs of text specifying which toolkit components to use when—to very advanced—configuration files supplied to a generic DSL generator.