Core Research and Development

Vision

To make Core accessible to a larger audience, building business information systems for an inter-networked environment.

To create one integrated development environment handling information model, information input/output including WEB user interface, report generation and integration with legacy systems.

General plan

1. Development of a complete language, called -S-language
2. Integrated development environment for this language, including UI tool
3. Code generation for target environments

Achievements

S-language adds formalization and several constructs to Core, such as complete type system, expression syntax including set algebra, syntax for describing interfaces (user interface, data import/export interface), event controlled expression evaluation, timers and processes. It also includes some advanced experimental constructs, such as sub-model inheritance.

First grammar definition and parser version was built in 1993-1995 using the meta programming environment from the Mjølner Project. Second version, was built in 2001-2005 in cooperation with the compiler team from Astrosoft in St. Petersburg, using an extended LEX/YACC system, generating an attributed object oriented AST in C++.

S-Language Tool (SLT) is a graphical programming environment combining graphical notation and textual representations of the s-language. It is built around the LEX/YACC system and AST. It includes the complete current version of the s-grammar and is used today for modeling and specification work. The internal AST can be saved in XML format for further processing.

A documentation generator and a partial translator to CoreBuilder was made in XSLT.

Current work

CoreWEB - It has been concluded that, given an S-model, it is possible to build (declare) a complete graphical user interface without programming effort, using and combining just 5 basic components. This method vastly reduces UI building efforts, introduces powerful reuse capabilities and at the same time allowing powerful extension to common UI concepts. First version of CoreWEB, built with XSLT and Javascript, has been finalized is currently running in Firefox browsers against the current production version of CoreCOM SOAP server. Version 2 of CoreWEB is in specification phase.

CoreBuilder code generation - Optimization of CoreBilder's code generation as well as rewrites for UNIX portability

Future work and research

Research

Review and refinement of S-language

Definition of complete attributed AST of the S-language, according to the Mjölner meta programming principles. Resulting AST should be available as XML-schema and in CoreBuilder format.

Parser, unparser and complete semantic attributing of AST. Possibly using JastAdd platform or XML/XSLT platform.

Runtime engine, directly running (interpreting) the AST.

CodeGeneration, such as Java, Java Byte Code, C++ ...

Optimizing execution:
- secure data-flow mechanisms in networked environments
- execution strategies such as caching, delta-evaluation
- dynamic execution strategies
- performance analyses
- support for massive parallelism.

Development

Building tools based on the attributed AST.
Tools would be, parsing, unparsing, documentation generation, code generation, visual AST editing tools such as modeling tool, UI building tool, report building tool, data import/export tool.
Building a multi-user development environment