

Open Code Translation from Executable UML Models

Erik Wedin

erik.wedin@dynamics.saab.se

-

Information Systems Development Department,
Software Systems (RTLIS)

-

© 2002 Saab Bofors Dynamics AB

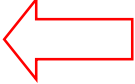
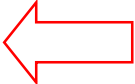
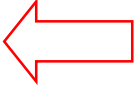
SAAB BOFORS DYNAMICS

INTERN
RTLIS Erik Wedin
2002-10-21 Sida 1



SAAB

Code Generation Techniques

- Model Based Development 
 - models with action semantics
- Closed code generation
 - the tool vendor is in complete control of how the resulting code performs and looks like
- Open code generation 
 - the user (i.e. Saab) is in complete control of how the resulting code performs and looks like
- Design by Elaboration
 - design/implementation information is added to the analysis model until the design is complete
- Design by Translation 
 - separates the analysis -"what" from the design -"how" via an architecture
 - the architecture contains the design information how the analysis models shall be translated to an implementation



Executable UML

- Executable UML is an upcoming standard, based on ongoing work within the OMG (Object Management Group, www.omg.org)
- Will be a well-defined profile (subset) of UML, extended with precise Action Semantics
- Possibility to develop SW and HW platform-independent specifications of the problem
- Supports the OMG Model-Driven Architecture (MDA) initiative
 - PIM - Platform-Independent Model - models the solution of a problem
 - PSM - Platform-Specific Model - models the details of the implementation
 - www.omg.org/mda
- Executable UML models = Executable Specifications
 - can be executed and simulated (platform independently)
 - can be translated to one/several implementation(s) onto one/several specific platform(s) - *without changing the models*



Executable UML - Fundamentals

- Subset of UML - extended with precise action semantics
- Provides guidelines for OOA and rules for consistency & completeness of executable UML models
- Separates the definition of the problem (OOA) from the definition of its implementation (a.k.a. RD-Recursive Design)
- Documented in the book “Executable UML - A Foundation For Model-Driven Architecture”, Mellor/Balcer 2002

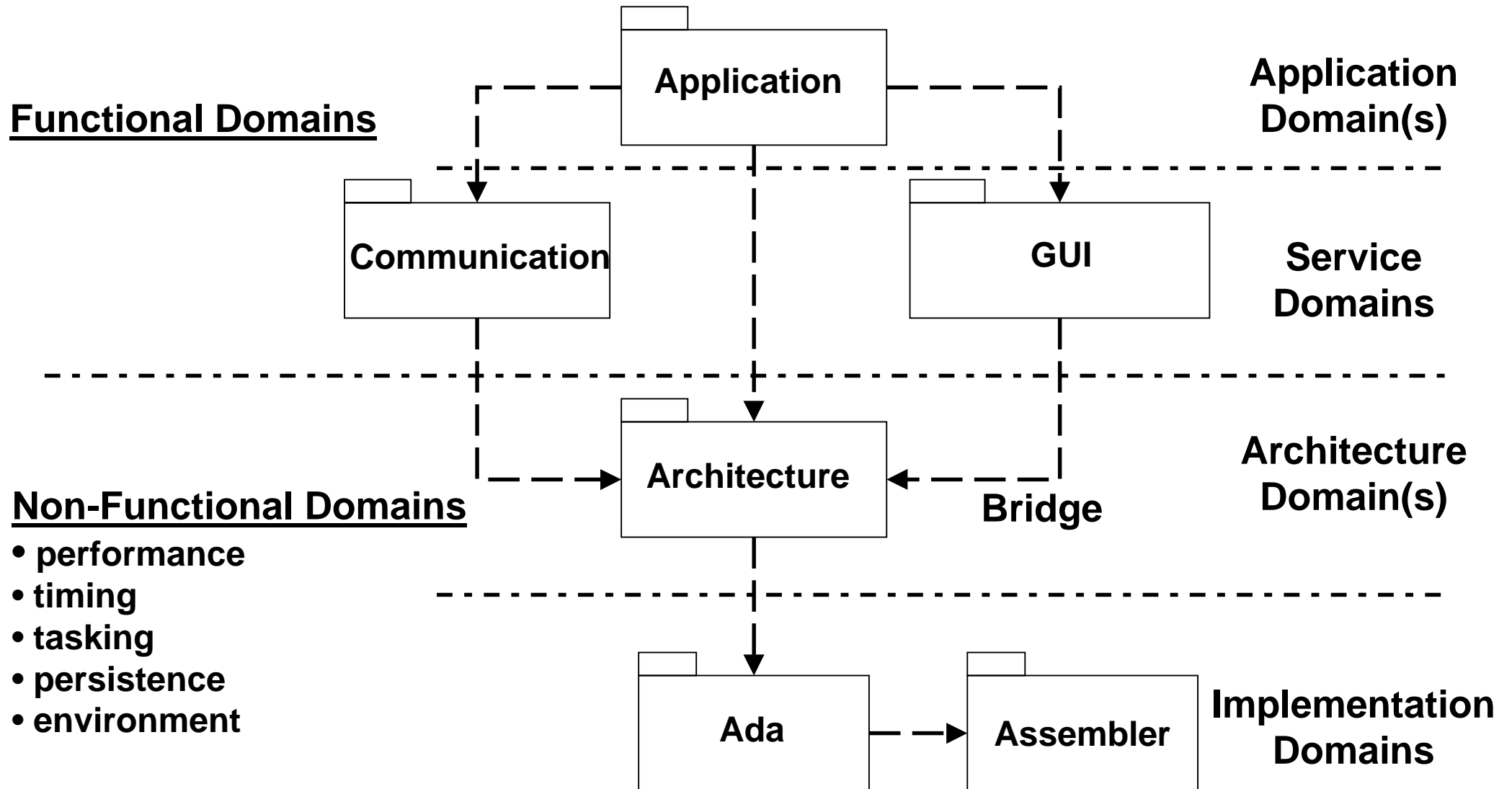


- Executable UML key areas
 - separation of systems into subject matters - Domains
 - each domain is either specified with an executable UML model *or* is a realised domain (i.e. provided as code)
 - domains are connected with dependencies - Bridges
 - translation of the models into an implementation with a Model Compiler

Domains and Bridges

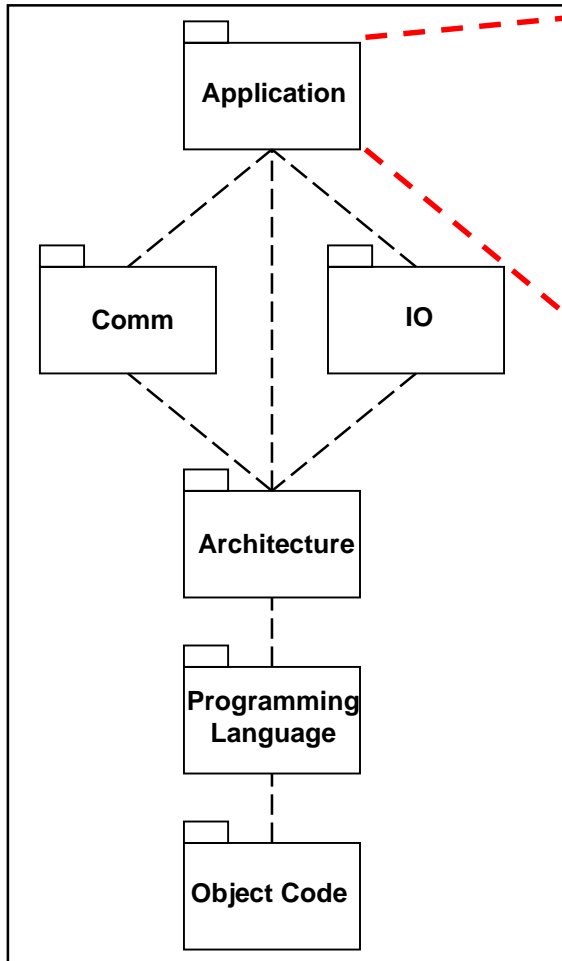
- Each Domain is a separate Subject Matter
 - subject matter \cong problem area
 - distinct objects, rules and policies
 - complete and independent of other domains
- Key goals
 - ability to re-use a domain without modification
 - ability to replace a domain with another that accomplishes the same mission
- Domains are connected with Bridges
 - a bridge is a dependency (it is not an API !)
 - a bridge contains mappings of concepts in one domain to concepts in another domain

Domain Types

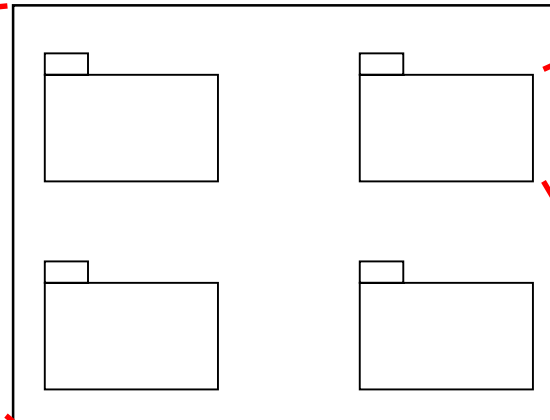


Executable UML - Main Diagrams

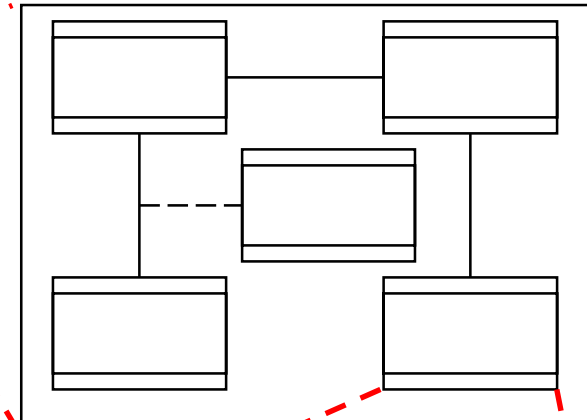
Domain Chart



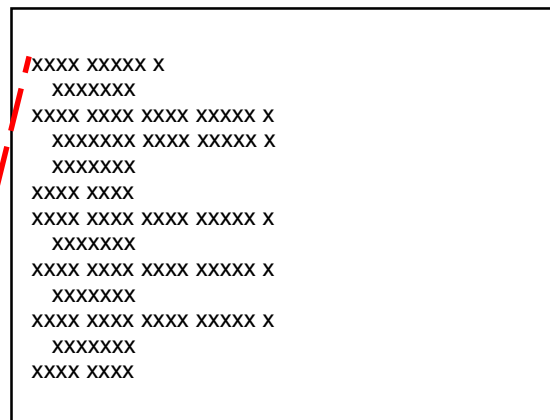
Domain Package Diagram



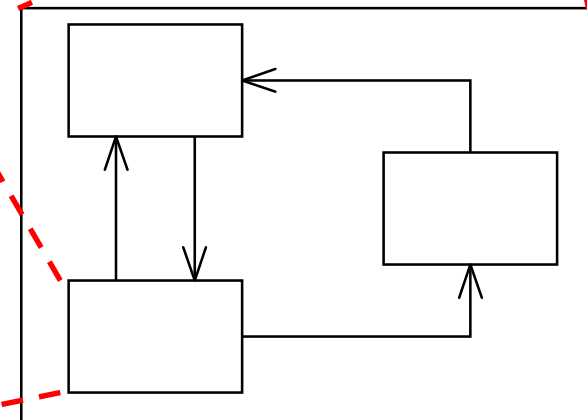
Class Diagram



Procedure

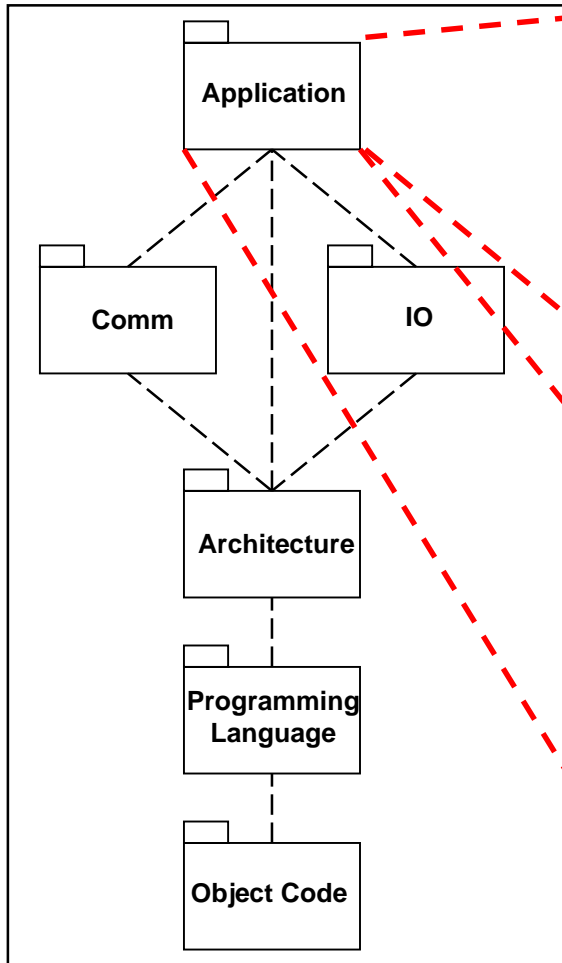


State Chart

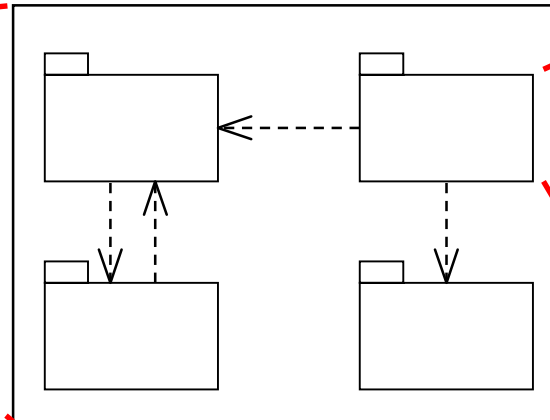


Executable UML - Derived Diagrams

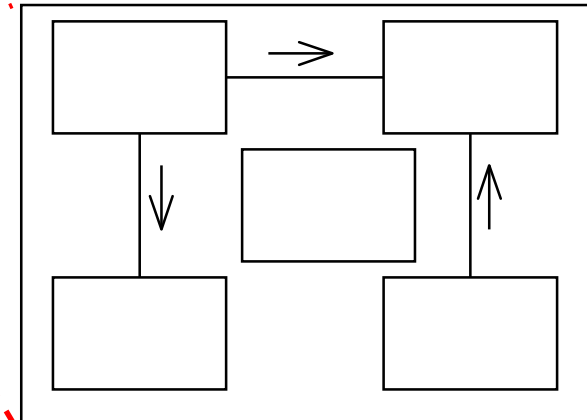
Domain Chart



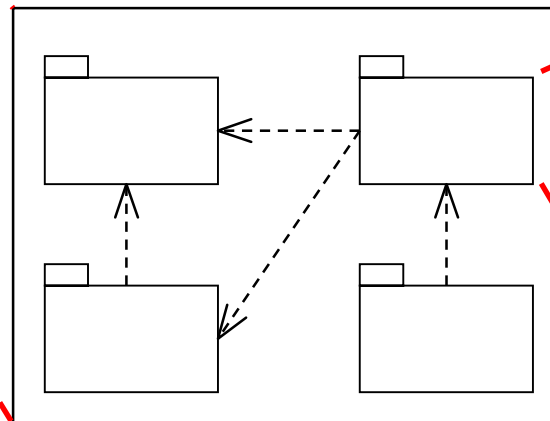
Package Dependency Diagram (Synchronous)



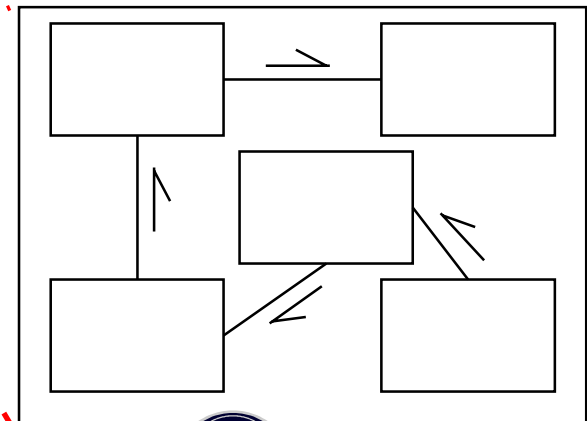
Object Collaboration Diagram (Synchronous)



Package Dependency Diagram (Asynchronous)



Object Collaboration Diagram (Asynchronous)



Software Architecture Domain

- The domain which deals with the design and implementation of the system
- Defines mapping rules between the UML models and the design and implementation patterns
- Prescribes how thread of control is to be initiated and controlled
- Prescribes how the code is to be structured and packaged

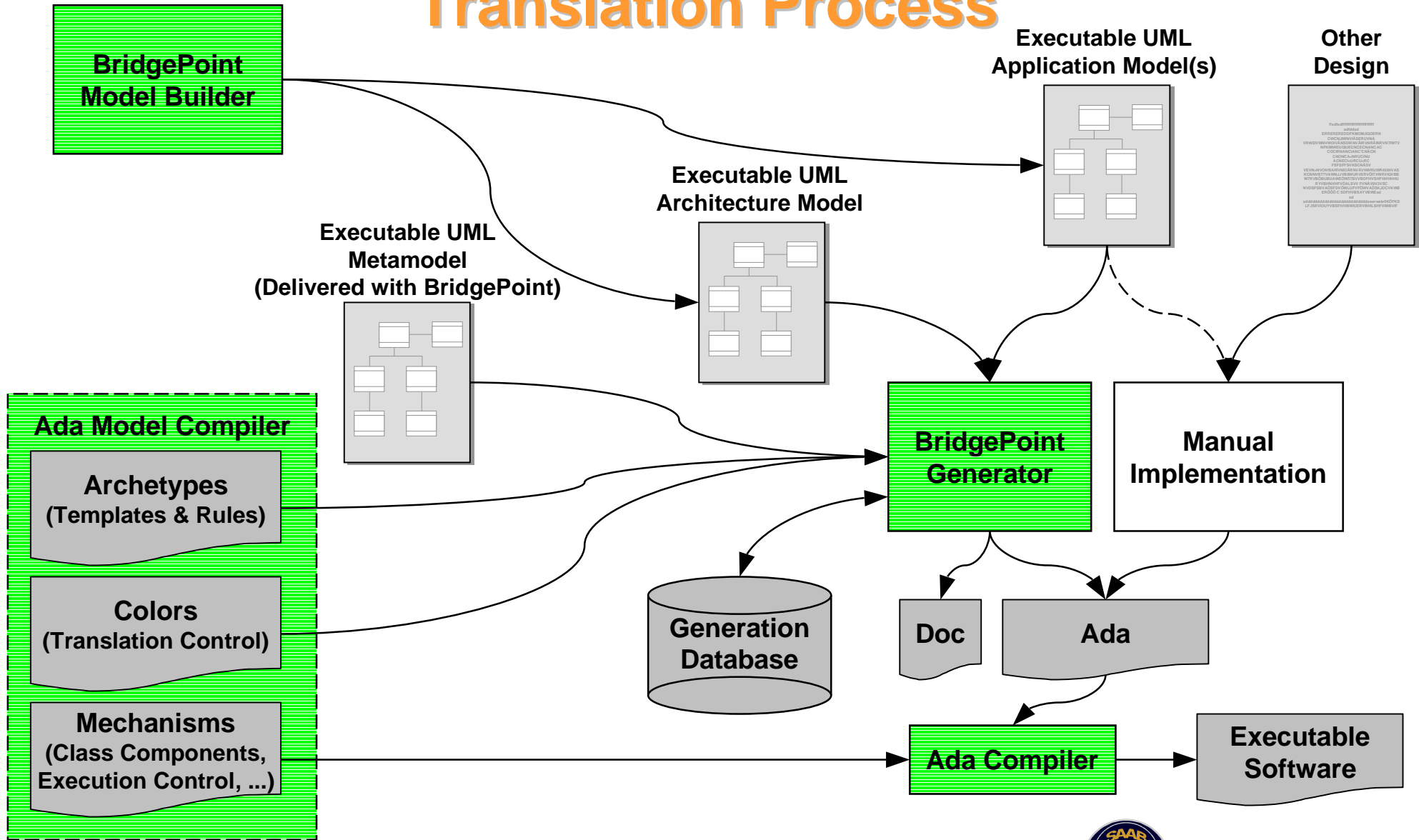
- This results in a uniform set of data, function and control structures that are used system-wide
- The Software Architecture is also called a Model Compiler

BridgePoint

- Tool-suite supporting Executable UML
 - Model Builder
 - executable UML model construction for each domain
 - Verifier
 - execution/simulation of Executable UML models
 - Generator
 - translation engine
 - Model Debugger
 - debugging of the translated system at model level
- Metamodel of Executable UML
 - a definition of how the UML model is stored in BridgePoint Model Builder
 - the Metamodel is also mentioned as the “OOA of OOA”
 - a Generation Database is created with the structure of the Metamodel
 - the Generation Database Metamodel is populated with information from the executable UML model
 - prerequisite for open code translation



Translation Process



SAAB BOFORS DYNAMICS

INTERN
RTLIS Erik Wedin
2002-10-21 Sida 11



SAAB

Model Compiler / Software Architecture

Main Components

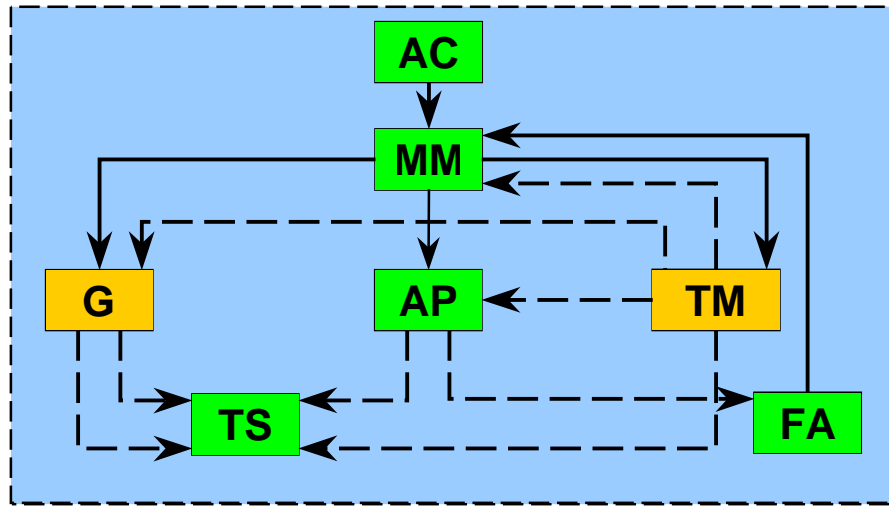
- Mechanisms
 - execution engine - event handlers, finite state machines, execution control, ...
 - platform independent
 - platform dependent (OS specific, HW specific, ...)
 - used system-wide
- Archetypes
 - combination of implementation patterns and translation rules
 - the rules fill out the patterns based on information from the UML models
- Colors
 - used to direct the translation to use different translation rules
 - injection of design decisions during the translation

Coloring Example

xUML-Model Classes

- AC = Aircraft
- AP = Autopilot
- TM = Telemetry Manager
- FA = Fin Actuator
- G = Guidance
- TS = Target Seeker
- MM = Mission Manager

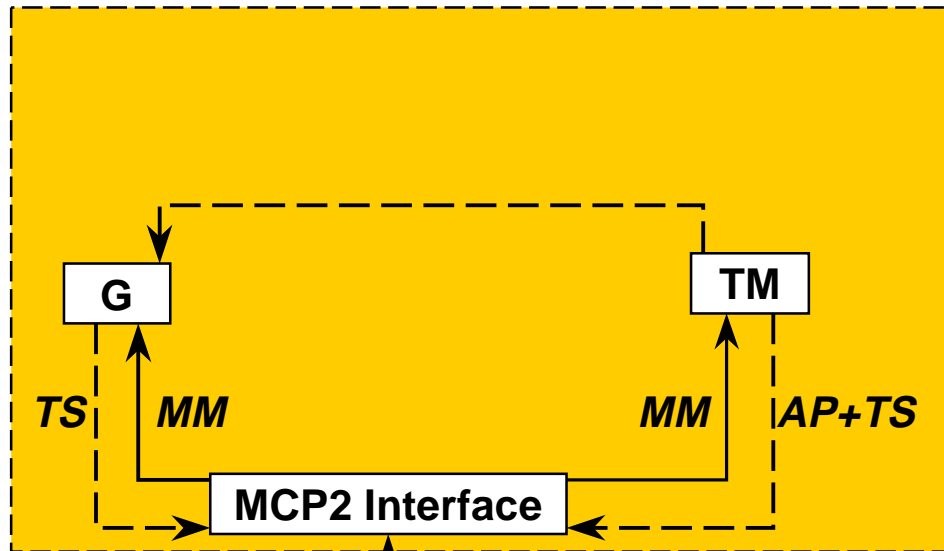
MC Executable UML Model



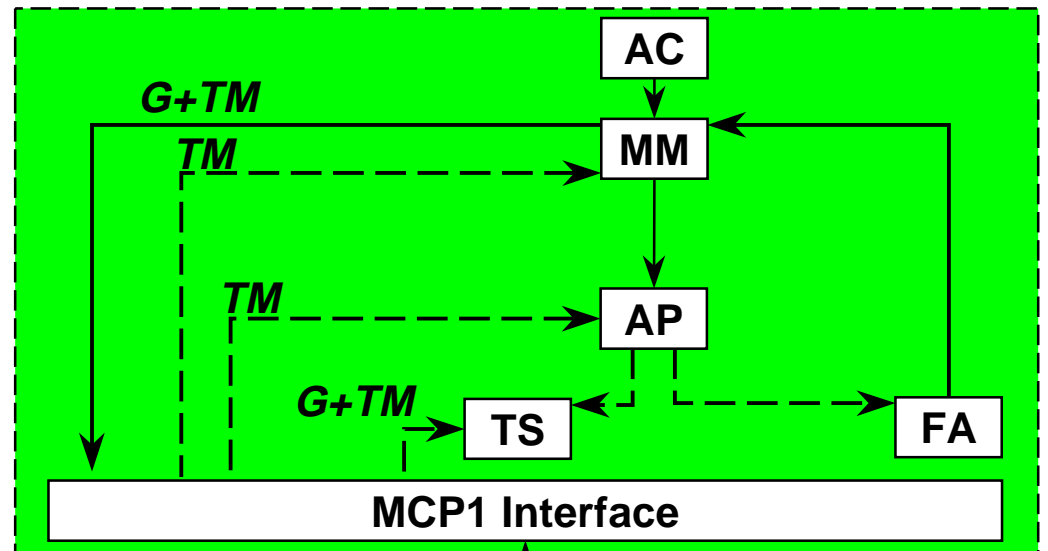
—————> = Event
 - - - - -> = Access

MC = Missile Computer
 MCP = MC Processor

MCP1 Software



MCP2 Software



SAAB BOFORS DYNAMICS

INTERN
 RTLIS Erik Wedin
 2002-10-21 Sida 13



SAAB

Model-Based Maintenance

- To address application behavior issues
 - Modify the relevant application model and regenerate
- To address performance-based issues
 - Modify the architecture/model compiler *and/or* architecture model and regenerate
- **DO NOT MODIFY GENERATED CODE!**



References



- Mellor/Balcer - “Executable UML - A Foundation for Model-Driven Architecture”

- the official book about Executable UML



- Leon Starr - “Executable UML, a case study”

- a hands-on example project

- Shlaer/Mellor - “Modelling the world in states” & “Modelling the world in data”

- books about Shlaer/Mellor Object-Oriented Analysis, predecessor to Executable UML

- Leon Starr - “Building Shlaer Mellor Models”

- still applicable for Executable UML models



References

- Project Technology - www.projtech.com
 - BridgePoint tools + DesignPoint (model compilers) tools + methodology
- Kennedy/Carter - www.kc.com
 - modeling tools, model compilers + methodology
- Kabira Inc. - www.kabira.com
 - systems and tools
- Rox Software - www.roxsoftware.com
 - model compilers
- Object Management Group - www.omg.com
 - UML + Executable UML + MDA-Model Driven Architecture
- SMUG2002 - <http://www.projtech.com/pubs/confs/2002.html>
 - Executable UML Conference in Tucson, Arizona, May 2002 hosted by Project Technology
 - Paper from Ericsson regarding Executable UML usage and Model Compiler development

Action Semantics Consortium

(www.kc.com/as_site/home.html)

- Members



- Supported by

