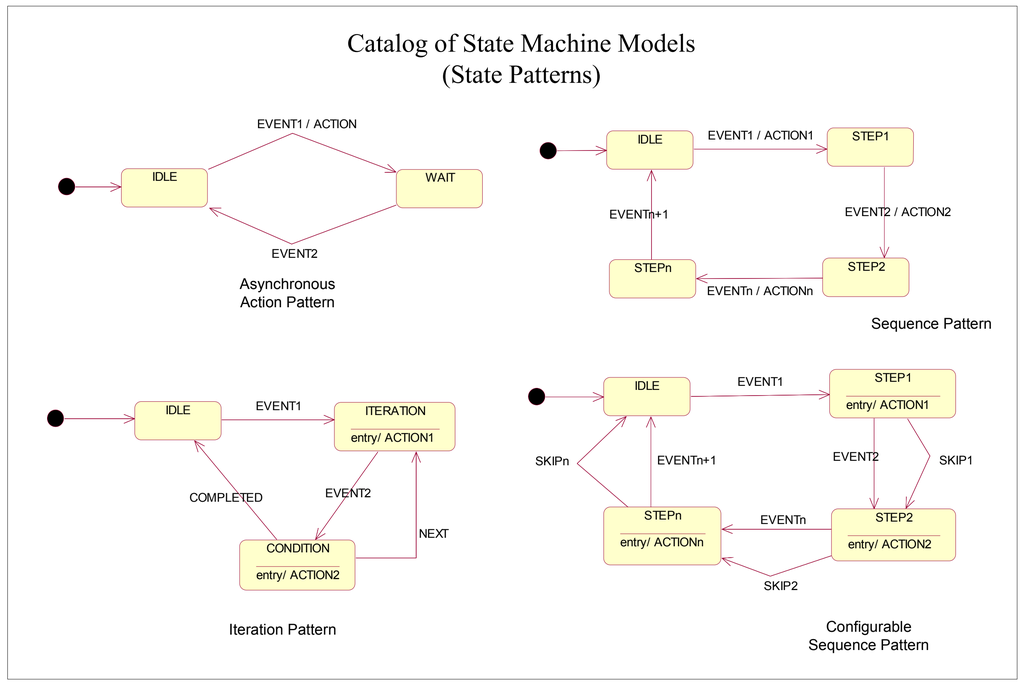
Workstation Software Framework
A Model Driven Development Framework

L. Andolfato, R. Karban
European Organisation for Astronomical Research in the Southern Hemisphere

PROCESS

The Workstation Software Framework (WSF) is a state machine model driven development toolkit designed to generate event driven software applications. State machine models are used to generate executables and it provides different, versatile generation options. It supports Mealy, Moore and hierarchical state machines. Generated code is readable and maintainable since it applies common design patterns such as the State and the Template design patterns. The framework promotes a development process that is based on model reusability through the creation of a catalog of state machine patterns.

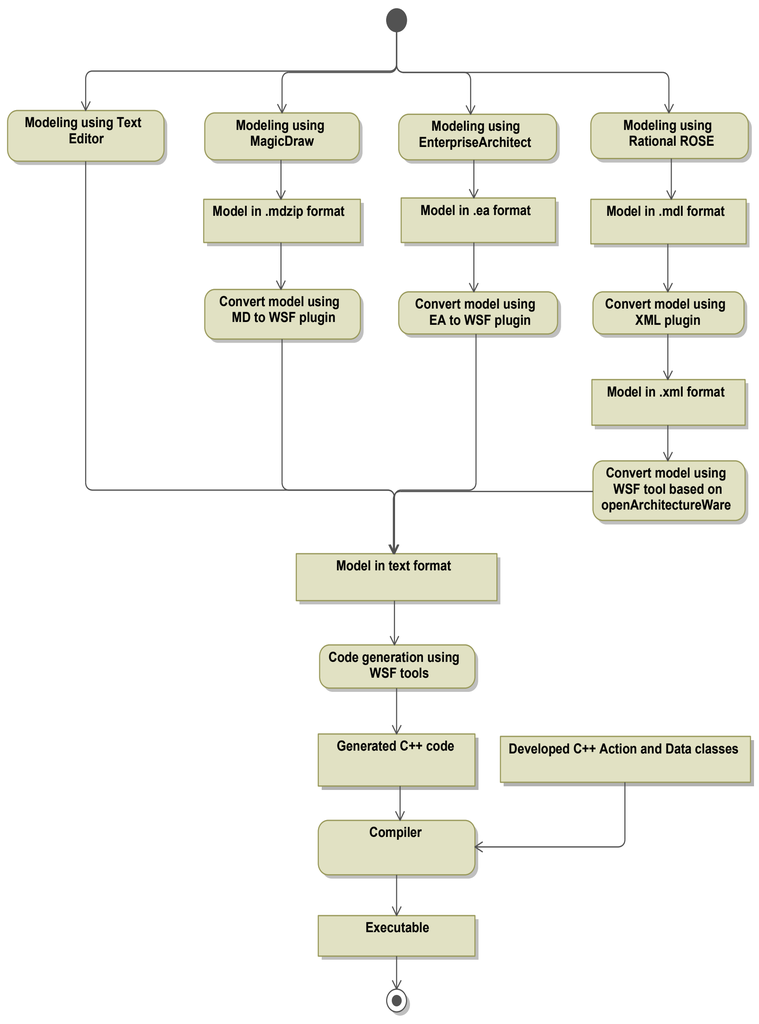


State machine models used to describe specific application behavior (such as sequences of commands, control loops, etc) can be completely reused in different applications simply by re-generating the code and adding a different implementation of the actions.

Reusability at model level has been found to be much more efficient (and easy to implement) than at class level since the interface between generated code and user defined components is well defined by the framework.

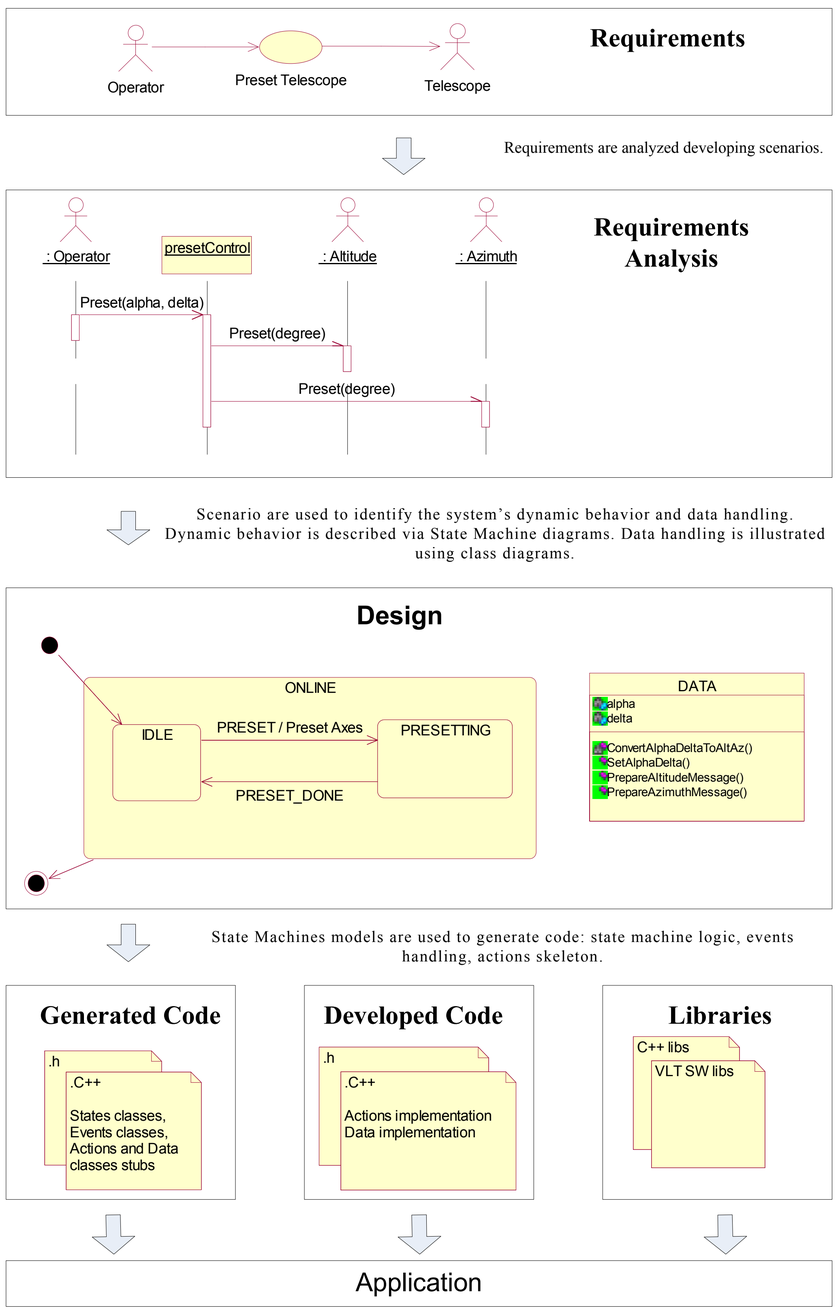
MODEL TRANSFORMATIONS

WSF provides the following model transformation functionalities:
- From textual representation to C++ classes
- From RationalROSE XML format to textual representation using XMI plug for RationalROSE and openArchitectureWare to convert XML model into textual representation
- From EnterpriseArchitect to textual representation via EA plug-in
- From MagicDraw to textual representation via MagicDraw plug-in



FEATURES

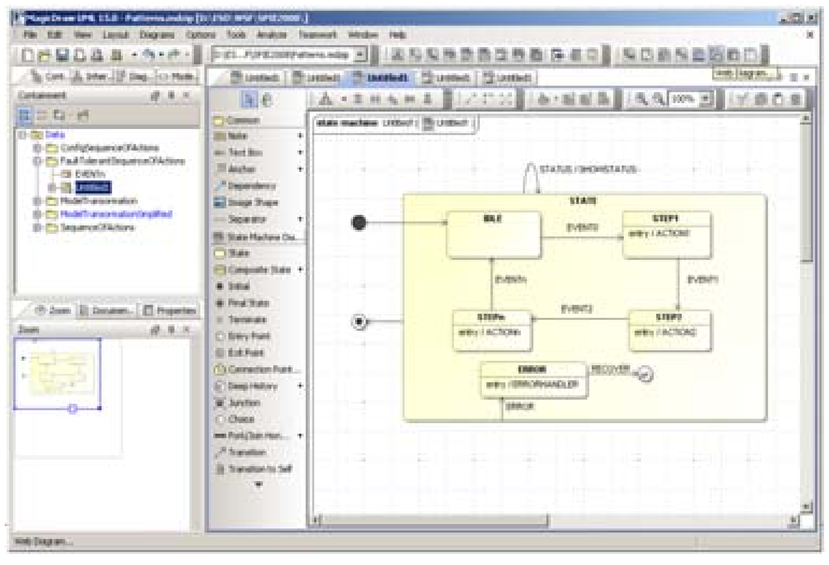
The Workstation Software Framework provides the following features:
- It supports Moore, Mealy and Haral state machine models
- It supports the History state
- It uses the actions classes concept to encapsulate more than one action per class



State Machine Models can be easily reused.

DEVELOPMENT

An application based on the Workstation Software Framework can be developed by:
- Adding manually states, events, actions and data classes. Since the architecture is based on design patterns, a WSF application can be built by extending the classes provided by the framework.
- Writing a text configuration file which describes the state machine model. A tool provided by WSF will transform the textual representation of the model in C++ state, event, and action classes. The developer has to complete the implementation of the action and data classes.
- Drawing the state machine model with Rational ROSE and export the model in XML format. A tool provided by WSF (based on openArchitectureWare) will transform the XML representation of the model in C++ state, event, and action classes. The developer has to complete the implementation of the action and data classes.
- Drawing the state machine model with EnterpriseArchitect or MagicDraw and run a plug-in to convert the model in textual representation. A tool provided by WSF will transform the textual representation of the model in C++ state, event, and action classes. The developer has to complete the implementation of the action and data classes.



SUPPORTED SOFTWARE PLATFORM

WSF has been developed for the Very Large Telescope Software platform based on:
- Linux
- GNU C++
- ESO VLT Software

ARCHITECTURE

The Workstation Software Framework (WSF) architecture is based on solid design patterns such as:
- the State design pattern to split state machine logic from action implementation
- the Template design pattern to hide event handling
- a modification of the Command design pattern for the action executions

Events (commands, replies, timeouts, database notifications, file I/O notifications, internal events) are propagated to the state machine logic which in turn will execute the associated action (transition actions or entry/exit actions).

STATISTICS ON CODE GENERATION

Project	Number of applications	Average number of lines of code per application	% Hand-crafted code per application
PRIMA	12	24004	21.24
APE	12	35020	25.08
NGC	4	16021	17.31
DL	1	24391	18.55

MAINTENANCE

An application based on the Workstation Software Framework can be maintained:
- By updating the model and regenerating states, transitions and events (without affecting data and action classes developed by the user)
- By fixing bugs on actions and data classes

FUTURE DEVELOPMENTS

Data classes used to access/store values, can be generated from a class diagram.

Code generation for the Action classes can be once the relation between actions and events and between actions and data is known.

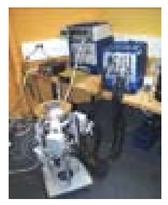
These relationships can be captured by a class diagram.
WSF code generation can be improved by adding a class diagram in the model. The class diagram should describe data classes, their attributes and the relation among data, event and action classes.

APPLICATIONS

Phase Referenced Imaging and Micro-arcsecond Astrometry for VLTI (PRIMA).

Active Phasing Experiment (APE).

Thermal Control for Auxiliary Telescopes (ATCS)



New General detector Controller (NGC)

Maintenance and Alignment of Delay Lines (DL)