Lockheed Martin (MDA SUCCESS STORY)

Tools used: Kennedy Carter's iCCG, UML, MDA

Description:
Lockheed Martin Aeronautics at Fort Worth, Texas has used the OMG's MDA to develop the F-16 Modular Mission Computer Application Software. Their goal was to achieve cross-platform compatibility and increased productivity and quality, all in the context of the demanding environment of avionics software development.

Problem:
The F-16 MMC team originally used traditional CASE tools with an OO modeling notation to specify the software before manually coding in Ada. When they migrated their development to Kennedy Carter's iUML tool, they gained the ability to use a UML action language which made their UML models executable. They could then test their UML models to verify their intended behavior before hand-coding the implementation.

More recently they have used Kennedy Carter's iCCG product to specify, in eXecutable UML, an Ada code generator which can automatically generate 100% of the Ada implementation.

By this means they guarantee that their UML models are entirely platform independent and portable across any future platform.

In Lockheed Martin's case, what constitutes the platform?

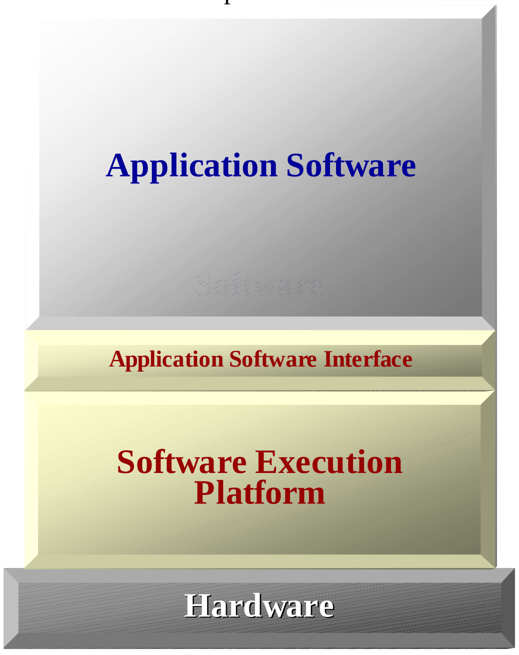

Figure 1 - F-16 Mission Software main architectural elements

SIDEBAR: Figure 1 shows the basis for the F-16 Mission Software architecture; it depicts the main architectural elements:

- **Software that is unique to the application(s) for which the embedded computer exists, it represents some 80-90% of the total software (in terms of long-term development cost);**
- **Application Software Interface, the boundary between the Application Software and the Software Execution Platform. This provides the methods by which the Application Software can make requests and use the services of the Software Execution Platform and the Software Execution Platform can provide its services to the Application Software;**

- **Software Execution Platform, low-level software, the purpose of which is to allow the Application Software to run on the hardware; The software execution platform incorporates device drivers, the built-in test and the RTOS.**
- **Hardware, the embedded system hardware for the F-16 Mission Management system.**

The software execution platform effectively raises the abstraction level of the hardware to provide a platform on which code generated from eXecutable UML models can run directly. Lockheed Martin's goal of complete cross-platform compatibility implies a very strong form of platform independence whereby the UML models which specify the Application software behavior can be ported without change even if the Application Software Interface changes.

Solution:
The use of MDA allows the mission software functionality to be formalized as eXecutable UML models (*xUML*); such models are Platform Independent Models (PIM) in MDA. Platform independence is essential if the goal of decoupling the models from any changes to the Software Execution Platform is to be achieved. We use the term *xMDA* to mean an MDA approach augmented by the use of xUML. xUML models are expressed using a UML action language based on the newly adopted precise action semantics for the UML (see http://www.omg.org/technology/documents/modeling_spec_catalog.htm#Action_Semantics).

Executable models support the MDA approach in two main ways. First, they allow early testing using simulation and debug tools. Secondly, since they are a full and formal specification of the system behaviour, they allow generation of the target code. Defining a mapping from the rigorously defined PIM (expressed in xUML) to the implementation is what is at the heart of xMDA. Since xUML models are executable and rigorous they act as much more than a simple visual agenda for the software developers, they actually embody all the business logic required to execute and verify the system.

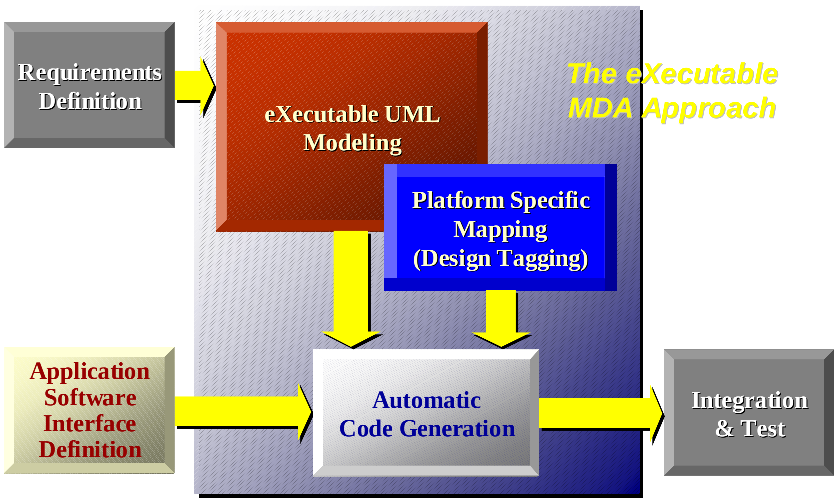
Requirements Definition
eXecutable UML Modeling
The eXecutable MDA Approach
Platform Specific Mapping (Design Tagging)
Application Software Interface Definition
Automatic Code Generation
Integration & Test

Figure 2 – Lockheed Martin's executable MDA (xMDA) process

Definition of the PIM to PSM Mapping

The basis of defining a mapping that allows full and automatic translation of the PIM models, expressed in xUML into the implementation is to build an xUML model of the translation system itself and execute it! The elegant conceptual coherence of employing xUML both in the business-modeling realm and in building the translator means that a minimum set of new skills has to be learned.

Figure 3 shows the basis for defining the mapping from PIMs (expressed in xUML) to their platform specific implementation. "Ordinary" analyst models (level 1) are used to populate the meta-model of xUML (level 2), which has all the necessary processing defined (remember it is an executable model in its own right) to populate a model of the implementation (level 3). This final model, which again is expressed in xUML, is executed to produce the implementation. The analyst models (level 1) are augmented with tags that act like "compiler directives" to the translation system and allow such facts as limited instance populations to be exploited in order to produce efficient target code.

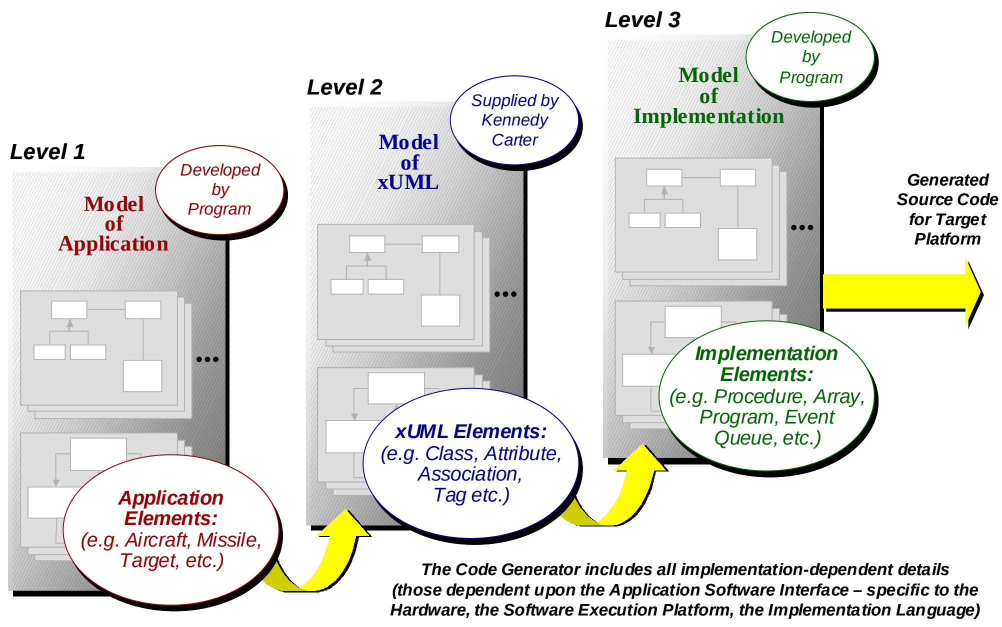
Level 1
Developed by Program
Model of Application
Application Elements: (e.g. Aircraft, Missile, Target, etc.)
Level 2
Supplied by Kennedy Carter
Model of xUML
xUML Elements: (e.g. Class, Attribute, Association, Tag etc.)
Level 3
Developed by Program
Model of Implementation
Implementation Elements: (e.g. Procedure, Array, Program, Event Queue, etc.)
Generated Source Code for Target Platform
The Code Generator includes all implementation-dependent details (those dependent upon the Application Software Interface – specific to the Hardware, the Software Execution Platform, the Implementation Language)

Figure 3 - Mapping PIM to PSM

Tool Support

There are two main areas where tool support is vital in order to reap the full benefits of the MDA process. First, iUML is used to provide the modeling and simulation environment for the xUML models. This tool provides dedicated intelligent support for xUML and allows models to be simulated on host and "debugged" visually at the UML level of abstraction. The second part of the tool-chain is the translation environment (level 2 in Figure 3 above) where application models are extracted from the iUML database and used to populate the translation engine. The translation engine is a specialization of the intelligent Configurable Code Generator (iCCG). iCCG allows developers to capture their mapping rules as xUML models and so produce any target implementation of which they can conceive. The specification of a mapping from PIM to PSM in eXecutable UML is itself highly reusable, allowing any set of application models to be generated onto the target.

Further details of these and other products that support xMDA may be obtained at www.kc.com.

Benefits:

The use of MDA with executable UML (xMDA) has provided many benefits to the F-16 project:

- The application models are expressed in a completely platform independent way and so can be reused across multiple hardware and software platforms;
- UML modelers are isolated from the software and hardware details and so can concentrate on a thorough exploration of the problem space;
- The hardware and software platforms can be upgraded without impacting the application models;
- Models can be tested at the earliest opportunity by executing them in the iUML Simulation environment;
- Rework is reduced with validated models;
- The mapping from PIM to PSM is specified in xUML with iCCG and is highly reusable;
- Code generation eliminates manual coding and eliminates the defects traditionally introduced in the coding phase;
- The xUML models are the primary source. Code is not maintained.

Taken altogether these MDA benefits have reduced application development time by 20% on the F-16 MMC program in addition to helping them achieve complete cross-platform compatibility.

Acknowledgments

This article was based upon a presentation prepared by:

Lauren E. Clark
Chief Engineer
F-16 Modular Mission Computer Software

Terry Ruthruff
Staff Specialist
Software Engineering Core

Bary D. Hogan
Methodology Lead
F-16 Modular Mission Computer Software

All of the Lockheed Martin Aeronautics Company