Meta Object Framework
Date: *July 4, 2014*
Originator: *Martin Beaver*

Architectural Decision (AD) – Meta Object Framework

At the ISA architecture meeting of Wed July 2, 2014 it was agreed that the current Andromeda modelling facility be adopted as a tactical solution. In fiscal year 2014/2015 work needs to be programmed to move this to a MOF 2.0 compliant repository.

Problem Context

MOF is a complementary infrastructure to component frameworks like J2EE, .NET and DCOM. It is an infrastructure dealing with a different aspect of application programming.

While the component frameworks like the J2EE architecture mask the complexity of the computing environment, they typically have no framework for masking the complexities of business models, business process, and business-logic. By abstracting away complexities of applications and systems, MOF is a framework for easing business model and process centric application development, as well as a foundation for tool integration.

The CSE Meta Object Facility will act as a central repository for all UML based business models. Platform specific models may be retrieved as either XML Schema or object representations. Calling applications and services will use these platform specific business models to generate and manipulate the standard SIGINT business objects (traffic, target, product etc.)

Functions

- Ability to load UML models from XMI or XML Schema files;
- For loosely coupled systems we require the ability to programmatically retrieve platform specific representations of loaded UML models as XML Schema;
- Models should allow for the mapping of structural elements to specific behaviour (services) via WSDL (see WSDL AD-43 for details on the selection of WSDL);
- Ability to support multiple versions of a model.

Business Requirements

- As SIGINT systems move from the current stovepipe architecture to a component–based one it is likely that there will be multiple programmatic consumers of SIGINT information. In order for this to succeed, each of these consumers must share the same understanding of the underlying business model.
- SIGINT exchanges information with partner agencies. In order for these external exchanges to take place a common information exchange model must be in place.
- The pace of change within the technologies covered by SIGINT is increasing. The hard coding of business models into the design of individual applications is too rigid to support this level of change in a cost effective manner. As a result an additional abstraction where models are captured centrally and used to influence the behaviour of applications is desired.

Risks

- While the underlying technologies (UML, MOF, XMI) are widely supported the level of maturity in the tools that implement these technologies is uneven;
- Moving to an increased reliance on centralized models rather than application specific code is a new model for CSE SIGINT.

- The ANDROMEDA project already has a metadata repository that resembles MOF. Moving to MOF will add considerable effort to this particular project.
- There will be a learning curve associated with adopting a MOF–based approach, and it will take time for projects to adopt this method. CSE's skills in the area of modelling and MOF–based development will need to be augmented for the full benefits to be realized in the future.

Motivation

The implementation of a centralized MOF based model system will provide many benefits to CSE.

Stakeholder List

- EPM01, EPM02, EPM03, EPM06, EPM07; CIO; ITPMO-ARCH

Constraints

- In order to ensure that the system can understand and properly represent structures, constraints and behaviors it is important that all CSE modelers use a standard notation.
- SIGINT models may be classified and have restricted dissemination rules. The Meta Object Facility must respect these rules and only allow authorized users / systems to view / update models.
- Behaviors in the system will be modeled via WSDL descriptors. These descriptors will allow a calling application to apply remote methods to the element.
- Modeling will occur in a UML design tool (e.g. Rational Rose)

Rationale

The options considered are detailed below.

Do Nothing (Models Continue to Reside in Application Code)

We could continue to model business objects in the applications themselves.

Dependencies

None.

Advantage:

1. No effort required to create a centralized MOF
2. No learning curve

Disadvantage:

1. Models must be recoded in each application that may access a particular service / component (increased development costs).
2. Very likely that these variances in models would lead to application integration issues.
3. No re–use.

Utilise Existing Andromeda Meta Object Repository

The Andromeda project has a similar solution that has been used for the past few months. It is possible that we could adapt this facility.

Dependencies

1. Acceptance of the Andromeda architecture.
2. Andromeda system must be brought to a production status by the end of 2003/2004.

Advantage

1. Reuse of existing code base would possibly reduce development time.
2. No changes required to Andromeda system to take advantage of MOF
3. Provides most of the advantages of a meta object framework (missing constraints and behaviour)

Disadvantage

1. This is a "non-standard" meta object system and so is not supported by COTS or GOTS (outside of the current Andromeda application) tools.
2. Currently the Andromeda system does not support behaviours or constraints; we would need to custom code a solution.
3. Currently the Andromeda system does not provide support for versioning of models
4. We will want to move to a standard based solution in the future. Therefore the selection of Andromeda as a tactical solution will have an impact on next years work plan.

Create a MOF Framework According to OMG MOF 1.4 Standards

This solution would se the creation of a centralized meta–object framework according to the OMG MOF 1.4 standard.

Dependencies

None.

Advantage

1. Standard, well supported technology set
2. Can be used with COTS modelling tools (e.g.: Rational Rose)
3. Provides all of the advantages of a meta object framework (including behaviour and constraints)
4. Support for Java Metadata Interface (JMI) specification (JSR-40)

Disadvantage

1. Standard does not include versioning of models. We would need to implement this.
2. The Andromeda project already has a metadata repository that resembles MOF. Moving to MOF will add considerable effort to this particular project.
3. Advent of UML / MOF 2.0 (expected within the next few months) renders 1.4 obsolete.

Create a MOF Framework According to Proposed OMG MOF 2.0 Standards

This solution would se the creation of a centralized meta–object framework according to the OMG MOF 2.0 standard.

Dependencies

None.

Advantage

1. Well positioned for advent of UML/MOF 2.0 compliant tools (likely to occur within next 12 months)
2. Provides all of the advantages of a meta object framework (including behaviour and constraints)

Disadvantage

1. MOF 2.0 is a draft standard and has not been fully approved. It is possible that last minute changes to the MOF 2.0 standard will render our implementation non-compliant.
2. No real vendor support (yet).
3. The Andromeda project already has a metadata repository that resembles MOF. Moving to MOF will add considerable effort to this particular project.
4. “Bleeding edge” implementation

Justification

- Immediate solution available (however it requires augmentation, specifically import/export facilities for XML Schema; behaviour)
- Delays a decision on full MOF compliance until the 2.0 standard is adopted by vendors.

Implications

- Work required in 2004/2005 to move to a MOF 2.0 compliant system
- Adoption of Andromeda technology implies Objectivity is accepted as a short–term standard.