OBJECT COMPUTING, INC.

Object Computing Inc. (OCI)

"Where the object is your success!"

Company Background

OCI (www.ociweb.com) has been providing innovative software engineering services to clients since 1992. Our focus is on supporting customer choice and openness by integrating heterogeneous systems. Our expertise is with multi-language, multi-platform mission critical infrastructure. For the last five years OCI has been providing commercial grade support and training for the leading open-source CORBA® products, (www.theaceorb.com) TAO and JacORB, and a C++ portability library called ACE.

By melding the combination of OCI training classes, open-source products and software engineering techniques OCI has been able to partner with clients to provide solutions that are both time and cost effective. Lately, by adding model-driven approaches to the design and delivery of projects we have been able to introduce new rigor, consistency, rapid evolution and increased reusability to client projects.

OCI's MDA Services

When delivering projects to clients OCI is committed to those software engineering techniques that ensure a successful outcome. One that is also commensurate with the client's comfort level and experience with differing software engineering processes. OOAD and eXtreme Programming are good examples.

An integral part of any OCI engagement is the use of training classes and mentoring to allow the client staff to understand and feel ownership of the solution.

Mentoring:

OO technology relies on the detection and understanding of patterns to solve recurring problems is an optimal manner. These patterns can be implemented at various levels of abstraction to enable their easy use by application developers. "OO Design Patterns and Frameworks" has been a frequently offered class to OCI clients to create awareness to this powerful strategy. Check here to see our offerings:- http://www.ociweb.com/education/oose.html. OCI leverages these patterns in its open source abstraction layer "ACE", its implementation of the OMG CORBA, and at the client domain level with frameworks.

Now with MDA we are offering a new one-day class for clients, "Introduction to Model Driven Architectures". This class is used to introduce clients to the overall approach so that during an engagement the clients understand the various steps being taken and how they conform to the emerging standards that will eventually populate the MDA domain.

OCI's MDA Consulting Experience:

OCI is already utilizing model-driven approaches for the development of domain-specific frameworks, languages, and associated tools for clients who require extensible, scalable, and flexible solutions. OCI's venture into MDA services is a natural extension of OCI's business and core competencies, which include:

- Software system architecture, design, and implementation using formal, object-oriented analysis and design methods including the development of models in UML using a variety of modeling tools.
- Software component framework design and implementation.
- Defining code-mapping rules, based upon formal models, and implementing code and artifact generators.

- Educating OCI's clients regarding formal object-oriented analysis and design methods including knowledge transfer through direct consultation and training sessions in a classroom setting.

OCI's staff includes architects and senior engineers possessing the skills and experience necessary to operate in an MDA environment including:

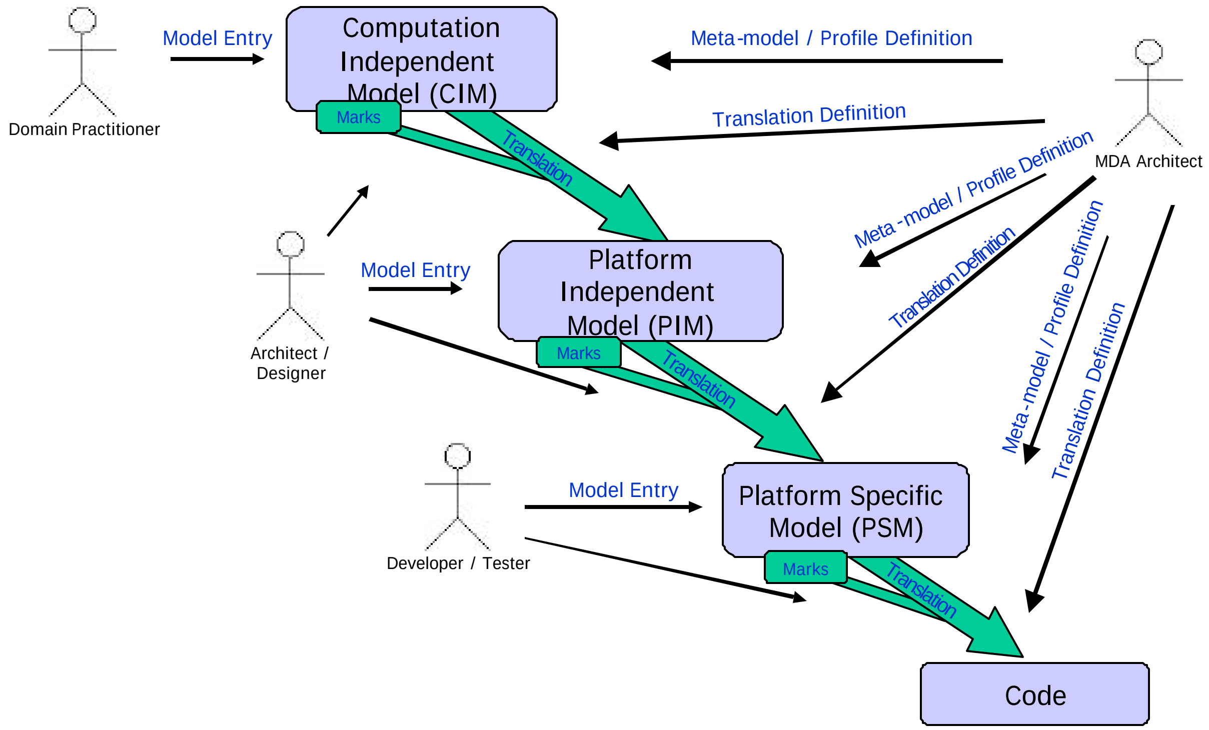
- Problem domain analysis and definition of abstract models as well as platform specific models.
- Identification and separation of "generalized problems," from which framework components arise, from "application specific features," which form the basis of a specific application.
- Application of software-design patterns and software-development idioms.
- Application of appropriate standard/reusable components and services.

Recent and continuing engagements include development of domain-specific frameworks, languages, and tools including code and artifact generators.

The OCI Team:

The figure below pictorially presents the top-level MDA process. Each of the actors in the diagram represents a role for members of the project. Initially, OCI will serve in the role of MDA Architect, while client resources will typically, but not necessarily, fulfill the remaining roles. In addition, OCI will prepare client staff to assume the role of MDA Architect.

> MDA Architect: Defines the meta-models and profiles for the Computation Independent Models (CIM), Platform Independent Models (PIM), and Platform Specific Models (PSM). The architect is also responsible for defining and creating the translations between the CIMs, PIMs, and PSMs.
>
> Domain Practitioner: Authors the Computation Independent Model to define the system.
>
> Architect / Designer: Applies "marks" that refine the translation process from Computation Independent Model to Platform Independent Model. This individual is also responsible for entering any additional Platform Independent Model elements that were not generated from the CIM to PIM translation.
>
> Developer / Tester: Defines any Platform Specific Model elements that were not generated from the PIM to PSM translation. Developers and testers are also responsible for adding any "marks" that refine the translation process from PSM to code.



In order to develop a project using the MDA approach it is necessary to first define the abstractions that will be used to represent the problem domain and the platform technologies that will be employed for system development. These abstractions will be formally described using meta-models or profiles. In some cases these meta-models or profiles may be realized by an existing OMG specification. The UML Profile for Enterprise Distributed Object Computing is an example of such a profile, which is used to define platform specific models. However, it is likely that profiles addressing problem-domain specific abstractions will be defined by the MDA architect.

The MDA Architect defines these models for use by one of the modeling roles (Domain Practitioner, Architect/Designer, Developer/Tester). The MDA Architect is responsible for using the tools available to capture these meta-models, which will provide the constraints for model entry. The MDA Architect will also define the translations between each of the models along with the appropriate marks that may be applied by the modeler to help direct the translation process.

MDA shifts emphasis away from authoring of code toward specification of system capabilities via several layers of models. Complete realization of a software system requires some handcrafted code, but over time dependence upon handcrafted code is reduced. The models used to specify a system are constrained and focused by the meta-models, which are defined by the MDA Architect. The meta-model takes the role of the system architecture. The meta-model is then refined by additional models each of which represent the system at decreasing levels of abstraction. Ideally, the goal of MDA is to reduce the number of intermediate models, but practicality may require additional intermediate models that provide the developer with an opportunity to influence the generated system.

Our belief is that no single tool will satisfy the needs of the entire project and that existing tools have yet to reach the necessary level of maturity and degree of interoperability. Moreover, it is essential that all models be represented via a common, standardized form, namely XMI. Consequently, OCI's role will most likely include the following:

- Integration of various tools, including possible development of intermediate translators, to achieve the requisite level of interoperability.
- Development of architecture modeling (meta-model) and domain modeling (CIM, PIM, and PSM) tools to fill capability gaps in the tool chain.

At the beginning of this engagement, OCI anticipates a discovery phase during which the MDA Architects will produce a high level architectural model, which identifies the system's primary use cases, and a detailed project plan to guide the effort.

Following this phase, the MDA Architects will perform domain analysis to refine the use cases and develop domain models. The MDA Architects will collaborate with the domain experts to insure that the models accurately represent the problem domain and relevant platform technologies. OCI will also guide the client through an evaluation of existing tools and provide input to "build or buy" decisions as needed and assist with the specification of requirements for interim tools developed specifically for this project.

Achieving the desired interoperability among the tools employed by this project will occur in several, incremental steps during early stages of the project. The MDA Architect will continue to refine and extend the meta-models, profiles and transformations. Additional tools will be developed as needed to fill the gaps in the existing tool set. Domain Practitioners, Architect/Designers, and Developer/Testers will use the project tool-set to author the various models. The results of this effort will validate and provide vital feedback regarding the quality of the meta-model elements.

The objective is a satisfactory implementation. Requirements can change over time. OCI's rapid development technique combined with frequent "drops" is designed to get hands on feedback, from users, as quickly as possible. Automated procedures following the MDA model enable fast cycle time and mitigate risk.