Interactive Objects

A case study using MDA according to the OMG with Arc Styler®

Jens-Hagen Syrbe

Jens-Hagen.Syrbe@io-software.com

Outline

- Problem setting
- The development process
  - Model-Driven Architecture at work
- The runtime environment
  - dvg's C++ component model
- Conclusions

Introduction: the history

- The company: dvg
  - IT center of German saving banks, approx. 1500 employees
  - Saving banks restructure to reduce cost without closing branches
  - Maintains 40 mio client records
- The old client management system
  - Written in assembler
  - Development continously since early 1970's
  - Platform: IBM OS/370 plus proprietary extensions to OS370, IMS hierarchical data base

Introduction: the project

- Task and constraints
  - Replace the client management system and open it to new channels (exact reimplementation plus some functional extensions)
  - The platform and runtime environment remains the same
  - Keep existing IMS structure and semantics, only extensions are allowed
- Facts about the project
  - Duration: 2 years
  - Team size: 7 developers
  - C++ source code: 11 MB
  - Approach based on Model-driven Architecture, using two models
    - business object model (~45 classes, ~200 associations)
    - data object model (~40 classes)

MDA: Business object modeling (1)

- Representation
  - Restricted UML subset to minimize complexity of model and code generation
  - UML class diagrams
  - Classes, single inheritance, stereotypes, attributes, operations,
  - Unidirectional associations, cardinalities
- Model as interface between business expert and developer
  - Dramatically improved quality of requirement analysis

MDA: Business object modeling (2)

- Model captures all information of business relevance
  - Persistence
    - Mapping to elements of data object model, no details of data base revealed at this level
  - Authorization level required to access model element
  - Integrity constraints (e.g. valid ranges of values or cardinalities)
  - Key attributes, ...
- Platform independent model
  - Platform meaning: implementation language and database
- ***Separation of business domain and technology***
  - Freedom to change database or implementation language without touching the business model

MDA: Data object modeling

7

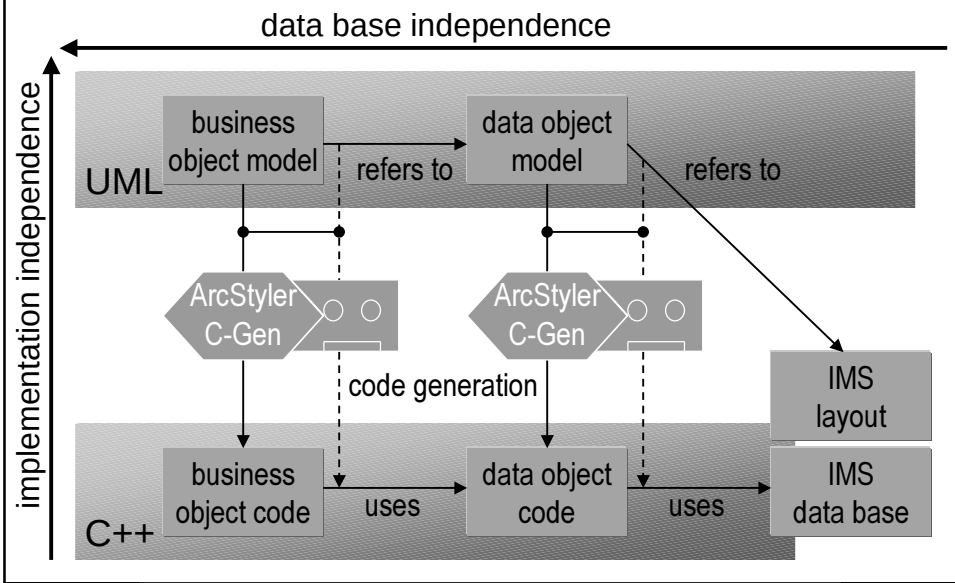
- Representation
  - Restricted UML subset to reflect data base structure
  - Classes, attributes
- Model captures all information necessary to find data and transfer data to and from the IMS data base
- Platform specific model
  - Platform meaning: data base
- Platform independent model
  - Platform meaning: implementation language

Interactive Objects

ArcStyler

MDA: The models and degrees of independence

8

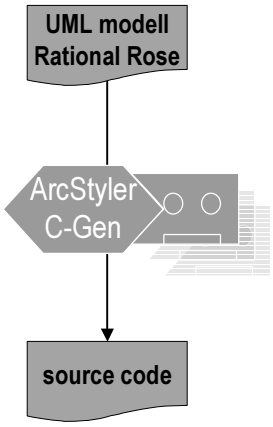


Interactive Objects

ArcStyler

MDA: Code generation

- ArcStyler C-Gen – a highly flexible code generator
  - programable by plugging in cartridges
  - cartridges:
    - sets of JPython-based template scripts
    - developed in special C-Gen IDE
- Code generation process
  - extract UML information from Rational Rose through COM-API
  - cartridge plugged into code generator operates on UML model
- 70% of overall code base generated
  - business object implementations
  - data object implementations
  - complete component infrastructure on top of plain C++

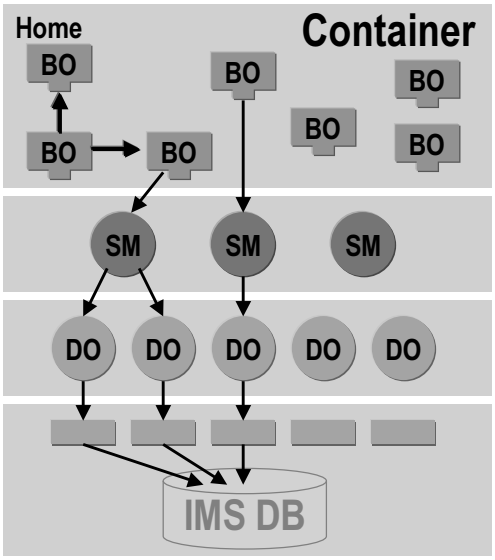


dvg's C++ component model

- Proprietary component model
  - Aligned to EJB entitiy beans with container-managed persistence
    - Home objects, home finder
    - Container with responsibility for object identity, caching, authorization, commit,...
    - Schema mapping from business-oriented data types to untyped bit fields in IMS segments
  - On top of plain C++
    - self-made design down to business-oriented basic data types
  - Consistent and performant implementation only feasible through high proportion of generated code

Business objects and data objects

11



identity
caching
commit

1-n BO-DO
schema mapping

1-1 DO-Segment
transformation of
basic data types

IMS segments

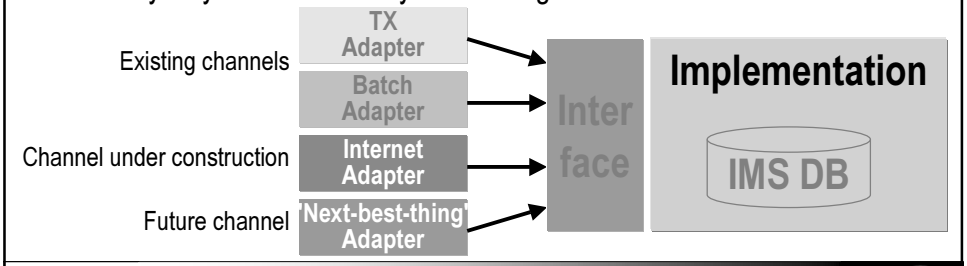
Interactive Objects

ArcStyler

Using the component model

12

- Adapter technology
  - Provide access to business objects in the client management system through well defined interface
  - Each adapter only translates requests from its notation to calls to business objects
  - All functionality for transaction control is implemented once inside the core of the client managemant system
  - Easy way to access the system through new channels



Interactive Objects

ArcStyler

Conclusions

- dvg leaped through 3 decades of software engineering from hacking assembler code to MDA
- The new system is online since February 2001
- Benefits of MDA gained through information-rich UML models and ArcStyler's code generator:
  - Increased quality of requirements
  - Increased quality and consistency of source code
  - Reduced development time
  - High flexibility and extensibility
- C++ leveraged by added value of component model
  - Made possible by flexibility of programmable code generator
    - Generated C++ source code consistent and optimized
    - Manual portions of C++ source code simplified

Interactive Objects

ArcStyler®

Interactive Objects

Please visit us in the demo area!

Model Driven Architecture for the Enterprise

ArcStyler®

http://www.ArcStyler.com/

Interactive Objects Software GmbH
2111 Bridgeway
Sausalito, CA 94965

Ph. +1-415-339 0412
Fax +1-415-339 0411

info@iO-Software.com