BLU AGE 2009 Edition
Agile Model Transformation

Model Driven Modernization
for Legacy Systems
Agenda

- BLU AGE Agile Model Transformation
  - Model transformation global overview
  - Enterprise application generation offering's

- BLU AGE Legacy Modernization
  - Reverse modeling for Legacy application
  - BLU AGE reverse modeling definitions, techniques and patterns
  - Case study: "Scafruit"

- BLU AGE Reverse Modeling principle
  - KDM as fundamental technique for reverse modeling
  - MDA and Ontology development
  - Knowledge base
  - Global overview of the transformation process

- BLU AGE Model Driven Modernization case study
  - Reverse modeling case study: Process and Steps
  - Reverse modeling case study: Demonstration
  - Workshop presentation: case study labs
BLU AGE Agile Model Transformation global overview

- BLUAGE is an MDA compliant software workbench Built-On Eclipse, that instantly transforms your Models into Agile Java EE and .NET business applications.


- Platform-Independent Models (PIMs) are free from technical concerns while Platform-Specific Models (PSMs) are annotated with platform-oriented configuration information to generate end users’ applications.

- PIM models are realized with available modeling market tools such as, Rational, Magic Draw, Enterprise Architect...
BLU AGE Agile Model Transformation: Build Edition

- The BLU AGE Build Edition is an integrated modeling environment built-on Eclipse to validate PIM models and generate in real time Enterprise Application

- Key features:
  - Model Driven Architecture support which transforms user friendly model elements into business application
  - Real time application generation & deployment within Eclipse or Eclipse-based IDE
  - UML2 model debugger (Executing the generated application & debugging it from model)
  - WYSIWYG Editor to map UI HTML mockups with models
  - Validation of input models by using OCL 2.x constraints that are related to the meta model classes.
BLU AGE Agile Model Transformation: Deliver Edition

- Integrated Generation Environment to fully generate your applications into the majority of existing and forthcoming frameworks and web-applications servers thanks to a large range of customizable transformation cartridges (BSPs)

- Key features:
  - Fully generate production applications compliant with your custom frameworks and architecture
  - Use 'standard' cartridges (BSP – BLU AGE Shared Plug-ins) for common enterprise architectures support
  - BLU AGE Software Factory to create and modify your own BSPs and improve generated application performance
  - Manage your projects, iteration and generation workflows through a single interface
  - Application delivery Life Cycle Management.

Generated Application
BLU AGE Agile Model Transformation: Screenshots

IBM Rational® RSM for PIM creation

BLU AGE Build Edition for PIM validation
Agenda

- BLU AGE Agile Model Transformation
  - Model transformation global overview
  - Enterprise application generation offering's

- BLU AGE Legacy Modernization
  - Reverse modeling for Legacy application
  - BLU AGE reverse modeling definitions, techniques and patterns
  - Case study: "Scafruit"

- BLU AGE Reverse Modeling principle
  - KDM as fundamental technique for reverse modeling
  - MDA and Ontology development
  - Knowledge base
  - Global overview of the transformation process

- BLU AGE Model Driven Modernization case study
  - Reverse modeling case study: Process and Steps
  - Reverse modeling case study: Demonstration
  - Workshop presentation: case study labs
Architecture Driven Modernization for Legacy Systems

- **What is a Legacy System?**
  - A piece of software that you have inherited, and is valuable to you

- **What are typical problems with legacy systems?**
  - Original developers not available
  - Outdated development methods used
  - Extensive patches and modifications have been made
  - Missing or outdated documentation

- **How to deal with Legacy?**
  - New or changing requirements will gradually degrade original design
  - ... unless extra development effort is spent to adapt the structure

- **What about Reverse Engineering?**
  - “Reverse Engineering” is the process of analyzing a subject system to identify the system’s components and their inter-relationships and create representations of the system in another form or at a higher level of abstraction.”
  - “Modernization” is the examination and alteration of a subject system to reconstitute it in a new form and the subsequent implementation of the new form.”
BLU AGE Legacy Modernization Edition

- BLU AGE Legacy Modernization Edition provides tools and methods with an integrated reverse engineering framework plugged into Eclipse.

- BLU AGE enables effective and efficient ways to treat models as first-class assets in reverse engineering process.

- BLU AGE has the ability to extract PIM models from existing legacy application in order to generate application source code conforming to new technical architecture design using Java EE or .NET without writing any line of code.

- BLU AGE is able to modernize legacy applications by software analysts, derive extracted PIMs into PSMs and link these PSMs to PDMs by generating set of transformations, while technical cartridges are designed by software architects.
BLU AGE Legacy Modernization principles

- BLU AGE provides a generic and extensible approach to model extraction and discovery from multiple types of legacy systems

- BLU AGE uses a model-based approach and a metamodel-driven methodology
  - Match different requirements systems modernization, data integration, etc
  - Use models operations and facilities: transformations, weavings, extractions, etc
  - Support methodology for defining extensions of the core metamodel and plug-ins to enable manipulating models, business rules, services, data, etc

BLU AGE Reverse Modeling: Global transformation process

- Legacy application
- Model Extraction
- PSM Model
- Architecture Driven Modernization
  - PIM Model
    - Full exploration support
    - Identify and convert all resources
    - Structure inventory and dependencies
    - Locate architectural aspects and layers
- MDA Model Transformation
  - PSM Models
    - Agile model transformation
    - App Source code generation
    - Java EE and.Net cartridges
    - Transaction and batch support
BLU AGE relies on standards and extensible frameworks

- BLU AGE provides a common tool box (discoverers, model transformation, model weaving, etc) based on generic and extensible frameworks and core metamodel (OMG™ specification)

- Apply common principles and frameworks supported by a wide user community to the extraction and further management of models from legacy systems

- Use as much as possible OMG™ standards and Eclipse™ projects in the modernization process
  - Eclipse™ Modeling Framework
  - ATL model-to-model transformation
  - AMW model-weaving
  - ...

© 2009 NETFFECTIVE TECHNOLOGY - ne peut être copié sans l'autorisation expresse de NETFFECTIVE TECHNOLOGY
The SCAFRUITS application case study is used by ‘Intermarche’ one of the main European companies in the retail sector.

This application is concerned with the provisioning of fruits and vegetables from referenced suppliers as well as the sale and distribution of these items to franchised outlets all over Europe. Shops interact by means of the SCAFRUITS application to have the best products at the best prices in the best conditions.

The application design and initial utilization start in 1994 using IBM VisualAge PACBASE (approach based on the PACBASE Access Facility (PAF), COBOL-like language respecting the VAP design guidelines). It has continuously evolved from this period. At this time, from a business viewpoint, the application is composed of 85 Transactional Processes (TPs) and 23 batch processes.

About its technical facets, the size of the application is estimated to be equal to 3M of LoC, 600 programs, 400 screens, 200 batch programs, 300 potential users, 48,000 product references with only 2,000 active references at a time. There are 350,000 transactions per day and 100,000 created order lines per day.

This case study project has a functional scope perimeter for product referencing and ordering including 15 TPs and 15 batch (business) processes restructured under the form of objects, in the OO sense and their links into instances of classes and associations in the PIMs outputs.
BLU AGE Legacy Modernization: Case study screenshots
Agenda

- BLU AGE Agile Model Transformation
  - Model transformation global overview
  - Enterprise application generation offering's

- BLU AGE Legacy Modernization
  - Reverse modeling for Legacy application
  - BLU AGE reverse modeling definitions, techniques and patterns
  - Case study: "Scafruit"

- BLU AGE Reverse Modeling principle
  - KDM as fundamental technique for reverse modeling
  - MDA and Ontology development
  - Knowledge base
  - Global overview of the transformation process

- BLU AGE Model Driven Modernization case study
  - Reverse modeling case study: Process and Steps
  - Reverse modeling case study: Demonstration
  - Workshop presentation: case study labs
KDM as fundamental technique for reverse modeling

- Knowledge Discovery Metamodel (KDM, OMG specification) used as intermediate representation data structure in the BLU AGE software modernization process.

- The goal of KDM is to ensure the construction of a collection of ‘concepts’ within a certain domain that can be also viewed as an Ontology. These ‘concepts’ are pieces of knowledge extracted and made explicit.

- KDM is used in the process of understanding and evolving existing software assets, IT portfolio management and software assurance.
Ontology to define concepts and explicit domain knowledge

- Ontology defines the common terms and concepts (meaning) used to describe and represent an area of knowledge.

- Ontology can range in expressivity from a Taxonomy (knowledge with minimal hierarchy or a parent/child structure), to a Thesaurus (words and synonyms), to a Conceptual Model (with more complex knowledge), to a Logical Theory (with very rich, complex, consistent and meaningful knowledge)

- Ontology contains:
  - Class,
  - Semantic,
  - Contract,
  - Transactional,
  - Wrapper,
  - Entity
MDA and Ontology development

- Ontology is used to separate implementations for each platform with explicit domain knowledge.

- This explicit knowledge can be used to reason about the extent of platform dependencies and the restructuring refinement transformations.

- Setting up ontology is used in MDA by applying platform-driven model for Automatic discovery of Legacy application, domain model, and platform dependencies management for technical architecture including Cobol mainframe, Enterprise Java, .NET, ...
Ontology Platform Models example

- Platform dependency constraint ontology description
- Platform instance ontology
- Generated platform ontology
Global overview of the transformation process

- BLU AGE provides tools that automatically process searching on large data volumes for patterns, that can be considered as data knowledge which involves understanding existing software artifacts
  - Classify taxonomy, create intersection classes for constraint sets and infer class hierarchy
  - Generate UML Profiles configuration against platform specification sort by platform constraint hierarchy and platform ontology
  - Provide configuration rules and annotated metaclasses with platform constraints

- BLU AGE also provides tools to chain together and combine model transformations in order to produce new transformations, and implement new operations on models
Agenda

■ BLU AGE Agile Model Transformation
  ▾ Model transformation global overview
  ▾ Enterprise application generation offering's

■ BLU AGE Legacy Modernization
  ▾ Reverse modeling for Legacy application
  ▾ BLU AGE reverse modeling definitions, techniques and patterns
  ▾ Case study: "Scafruit"

■ BLU AGE Reverse Modeling principle
  ▾ KDM as fundamental technique for reverse modeling
  ▾ MDA and Ontology development
  ▾ Knowledge base
  ▾ Global overview of the transformation process

■ BLU AGE Model Driven Modernization case study
  ▾ Reverse modeling case study: Process and Steps
  ▾ Reverse modeling case study: Demonstration
  ▾ Workshop presentation: case study labs
The overall goal of the case study modernization is to replace existing application by a renewed application based on Java EE.

The modernization process is based on the implementation principles and details of the reverse components, with the generation of text based on a semi-natural language, instance of metamodel dedicated to the way to use VisualAge PACBASE (VAP) using ATL model transformation.
Reverse modeling case study: transformation process

- **VAP Extraction**
  - XML Files
  - Transactional Processes (TP)
  - Batch Processes
  - Dialogues

- **Discovery phase**
  - Facing complex system, the best strategy is to divide and transform it into a set of models which can be homogeneously handled.
Reverse modeling case study: transformation workflow

- From the COBOL legacy programs we extract XML files representing 100% of the contained information (with comments)

- The total discovery and understanding method may be quite expensive in time of execution and in space to store resulting models

- The solution is to discover only part of the COBOL program (Data structure alone, Control structure alone...) involving human experts and highly rely on different kinds of ATL transformations
  
  - Macro TP Model extraction
  - Micro TP Model extraction
  - Annotated Macro & Micro TP Model
Reverse modeling case study: transformation workflow

Macro TP view model and annotated Macro TP model

Micro TP view model and annotated Micro TP model
Reverse modeling case study: transformation workflow

Global TP view model

Final BLU AGE PIM extracted model
Reverse modeling case study: transformation steps 1/3

- The technical approach is based on the PACBASE Access Facility (PAF) component of VAP

- The Reverse modeling process relies on predefined metamodels starting from Ecore. The PAF tables’ structure is thus represented as an instance of the Ecore metalanguage.

- Once the complex organization of VAP is captured, PAF logic’s metatypes are members of this metamodel. By definition, the Table metatype for instance conforms to the EClass Ecore metatype.
The case study table structure is an instance of the Table metatype at the PAF Table’s structure level. PAF is thus used to populate records PAF tables.

PAF Queries Extractions are composed of several XML files. Several ATL transformations are next run in sequence to re-create, explicitly, the dependencies between VAP entities.

The PAF persistence model is computed by means of an ATL transformation from the PAF extraction model. The same applies for the PAF associations.
The PAF associations model complements the PAF persistence model by supplying reverse navigability.

The reverse engineering method is in fact divided into three phases:

- The populating phase contributes to have rich cartographic view of an VAP application based on the transformation processes.
- The interpretation phase which to parse the COBOL-like code conforms to predefined metamodel.
- The publishing phase based on VAP models and BLU AGE models.

VAP-oriented COBOL:

```cobol
SousFonction 'FERMUTR DU FICHIER RPT293WL' PG08FPA { 
  IF (Z499-XCPF02A = SPACES AN Z499-XCPF07A = SPACES) { 
    MOVE 5-PAT0-CPTEUNR Z499-XCHR29 
    MOVE 'LUS' Z499-XO7XCO 
    MOVE 'RPT293WL' Z499-XO6XI0 
    MOVE 'FI' Z499-XCPF02A 
    PERFORM FIA 
    MOVE SPACES Z499-XCPF02A 
    CLOSE PA-FICHIER 
  } 
}
```
Reverse modeling case study: transformation project structure

BLU AGE Reverse modeling project for one transaction process

BLU AGE Reverse modeling Meta models used for the transformations processes

BLU AGE generated PIM models for Java EE publication
Reverse modeling case study: transformation details

BLU AGE extraction workflow

BLU AGE Interpretation workflow

BLU AGE Representation workflow
For more information: http://www.bluage.com