Introducing BEASI™ 3.0

Biner Enterprise Architecture Standard Integration

A Biner software plugin for Sparx Enterprise Architect
The concept is based on the union of three global standards:

- **TOGAF 9.1**
  as the established framework and method to develop the architecture

- **ArchiMate 2**
  as the unified notation to describe the architecture

- **Sparx Enterprise Architect**
  as the modeling tool and repository to document and share the architecture
**Benefits**

**Effective EA**
By uniting popular global standards your organization can benefit from shorter lead times, increased productivity and reduced business risks.

**Properly Developed Structures**
Access to the TOGAF 9.1 method and the notation ArchiMate 2.1, gives you a developed structures for your projects, giving you a kick start.

**Reuse of Information**
The integration of method, notation and tool makes your projects structured with smooth management. All the information you create can be reused.

**Cross Project Collaboration**
A central repository makes it possible to share all information you create and enables you to run several change projects simultaneously.
Features

Highlights

- Automatic generation of ArchiMate 2 compliant viewpoints
  - Choose one element and BEASI does the rest for you

- Improved relationship matrix
  - Showing all derived relations, several steps away
  - Make a derived relation into an explicit relation
  - Enhanced help when adding new relations

- Automated relations management
  - Diagrams can be checked for nested elements
  - BEASI helps you choose the correct relation
  - Hide redundant relations on a diagram

In order to further improve the support for executing architecture using TOGAF, ArchiMate and Sparx Enterprise Architect, Biner have released BEASI version 3.
Tools for ADM execution

Preliminary and Vision Phases (P & A)
- Building a repository and project structure

Architecture Development Phases (B, C & D)
- Generation of gap diagrams
- Consolidation of gaps

Architecture Planning Phases (E & F)
- Identification and analysis of gap dependencies
- Association of gaps to work packages (projects), deliverables, plateaus (transitions)
- Generation of Architecture Definition Increments Table (i.e. a draft project structure)
Getting started

Prerequisites
- Ensure Sparx Enterprise Architect is installed
- BEASI is installed

Execution
1. Start Sparx Enterprise Architect and enter your license key in the dialog that appears
2. Once Sparx Enterprise Architect starts start a new project and save it
3. In the wizard dialog that opens automatically locate the Biner EASI technology
4. Select the BEASI tutorial and select ‘Ok’
5. Go to the project browser and locate the diagram BEASI tutorial

Result
- BEASI successfully installed with tutorial
Building a new structure

Prerequisites

Ensure the standard version of BEASI is installed
(In the trial version it is only possible to view an export of the template structures)

Execution

1. Open an existing project or create a new one (CTRL+N)
2. Select the **New model from pattern** icon in the project browser
3. Locate the **Biner EASI technology** in the dialogue box
4. Select one (or more) of the following structures:
   - BEASI ADM template
   - BEASI Repository template
   - BEASI Repository and ADM example
5. Select ‘Ok’ and wait for the structure to be created
   (this can take a while)

Result

The selected structures are created in the project browser
Building models

Prerequisites
A package structure have been created in the project browser

Execution
1. Modify and amend template diagrams included in the predefined structure as required
2. When additional views are needed, create a package to contain these by selecting the **New package** icon 📑 in the project browser
3. Insert a new diagram by selecting the **New diagram** icon 📑 in the project browser
4. Select the appropriate BEASI viewpoint in the dialog box that appears

Result
A diagram based on ArchiMate viewpoint is created
Relationship Matrix

1. Showing all derived relations, several steps away

2. Make a derived relation into an explicit relation

3. Enhanced help when adding new relations
Gap analysis

Prerequisites
1. Create a viewpoint package (e.g. an Application landscape)
2. Add a baseline package (containing the word baseline) and build a baseline view
3. Add a target package (containing the word target) and build a target view

Execution
1. In the project browser select the viewpoint package
2. Select Gap analysis from the Extensions > BEASI menu

Result
A gap diagram is created, where differences between baseline and target are highlighted
Gap consolidation

Prerequisites
- Gap elements have been added to all gap diagrams describing the required changes

Execution
1. In the project browser select top package of the architecture definition project
2. Select **Generate Consolidated Gap Diagram** from the Extensions > BEASI menu

Result
- All gap elements are identified in the project and presented on a single diagram
Deriving gap dependencies

Prerequisites
A consolidated gap diagram has been created

Execution
1. Open the consolidated gap diagram
2. Select Derive gap dependencies from the Extensions > BEASI menu

Result
- Derived relationships between gaps are added to the diagram.
- The elements behind the derived relationships can be visualized using the command Show elements on gap dependency path
Migration planning

Prerequisites

- All gaps are identified and analyzed

Execution

1. Place a group of previously defined gap elements on a new diagram
2. Select one of the following alternatives from the Extensions > BEASI menu:
   - Connect to new work package
   - Connect to new plateau
   - Connect to new deliverable

Result

Depending on the alternative a new work package, plateau or deliverable is created associated with the previously selected group
Migration planning

Prerequisites

- This feature generates a draft project plan in the form of a TOGAF architecture Definition Increments Table
- In order to allow generation gaps must have been associated to work packages and the gaps must have been associated with a transaction architecture represented by a plateau element.

Execution

1. In the project browser select the top package of the ADM project
2. Select **Generation of Architecture Definition Increments Table** from the Extensions > BEASI menu
3. Specify a filename to save the plan

Result

This feature generates a csv-file which can be imported to excel
Generating documentation

**Prerequisites**
This feature utilizes the built in virtual documentation functionality in Sparx Enterprise Architect

**Execution**
1. In the project browser locate the predefined ADM Deliverables package
2. Select one of the predefined master documents e.g. Architecture Definition Document
3. Select **Generate Documentation RTF/PDF** from the Project > Documentation menu

**Result**
This feature generates a RTF document using the master templates in Sparx Enterprise Architect
If you have not yet downloaded BEASl or want more information please visit

http://www.beasisoftware.com