Enterprise Architect

User Guide Series

# Domain Models

What support for modeling domains? Sparx Systems Enterprise Architect supports a range of modeling languages, technologies and methods that can be used in modeling and integrating different domains to reflect aspects of a complete business or process.

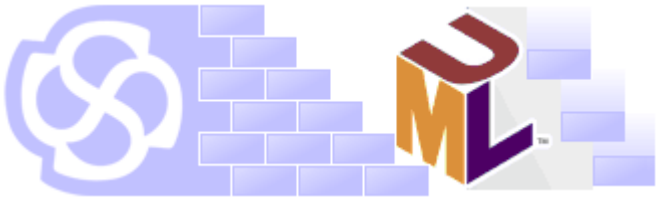
Author: Sparx Systems

Date: 7/08/2019

Version: 1.0

# Table of Contents

# Domain Models



Enterprise Architect provides support for a rich range of modeling languages, technologies and methods, most of which have been built as profiles or by the use of stereotyped elements that extend the basic UML elements. The power of modeling is the ability to integrate the various representations of a system, and stitch together models from a variety of domains and disciplines. For example, the ability to integrate models that describe the geospatial aspects of a feature in the world such as an airport, with regulatory and air traffic control models and baggage handling system models, provides a clarity that has not been possible before. It is the ability to model these concepts in the language of multiple disciplines, and then to tie them together in a single modeling environment, that makes Enterprise Architect such a useful and productive tool.

## Modeling Domains

| Domain | Description |
|---|---|
| ODM | Enterprise Architect enables you to develop large-scale ontologies within the fully-integrated modeling environment, for your project domain.<br>ODM helps you to develop a formalized representation of business semantics and taxonomies, and a knowledge representation based on those formalizations. |
| Requirements | Enterprise Architect is one of the few UML tools that integrate Requirements Management with other software development disciplines in the core product, by defining requirements within the model. |
| Business Modeling | Modeling the business process is an essential part of any software development process, enabling the analyst to capture the broad outline and procedures that govern what it is a business does. |
| Business Rules | Business Rule modeling captures the rules that govern a business, and their relationships with the entities and specific tasks within the organization or system. |
| BPMN | The Business Process Model and Notation is specifically targeted at the business modeling community and has a direct mapping to UML through BPMN Profiles; these profiles enable you to develop BPMN diagrams quickly and simply. |
| BPEL | Business Process Execution Language is an executable language for specifying interactions with Web Services.<br>Enterprise Architect uses the BPMN profile as a graphical front-end to capture BPEL Process descriptions. |
| SysML | SysML is a general-purpose graphical modeling language for specifying, analyzing, designing, and verifying complex systems that might include hardware, software, information, personnel, procedures and facilities. |
| Data Modeling | Enterprise Architect provides easy-to-use tools for building and maintaining all of the fundamental data models - Conceptual, Logical and Physical; because |

| | |
|---|---|
| | Enterprise Architect lets you visualize each type of data model in the same repository, you can easily manage dependencies between each level of abstraction. |
| XSD | Enterprise Architect supports rapid modeling, forward engineering and reverse engineering of W3C XML schemas (XSD), critical for the development of a complete Service Oriented Architecture (SOA). |
| WSDL | Enterprise Architect enables rapid modeling, forward engineering and reverse engineering of W3C XML Web Service Definition Language (WSDL), critical for the development of a complete Service Oriented Architecture (SOA). |
| SPEM | The Software and Systems Process Engineering Metamodel (SPEM) is a conceptual framework for modeling, documenting, presenting, managing, interchanging, and enacting development methods and processes.<br>SPEM 2.0 focuses on providing the additional information structures that you require for processes modeled with UML 2 Activities or BPMN/BPDM. |
| ArchiMate | ArchiMate is an open-standard enterprise architecture language based on the IEEE 1471 standard, providing a common language for describing the construction and operation of business processes, organizational structures, information flows, IT systems and technical infrastructure.<br>It enables Enterprise Architects to clearly describe, analyze and visualize the relationships amongst business domains. |
| ArcGIS | ArcGIS is a suite of Geographic Information Systems (GIS) software products developed by Esri. |
| AML | The Archetype Modeling Language (AML) defines a standard means for representing clinical information. |
| Data Flow Diagrams | A Data Flow diagram (DFD) is a graphical representation of the flow of data through an information system; it can also be used to visualize data processing (structured design).<br>Developing a DFD helps in identifying the transaction data in the data model. |
| Entity Relationship Diagrams | Entity-relationship modeling is an abstract and conceptual database modeling method, used to produce a schema or semantic data model of, for example, a relational database and its requirements, visualized in Entity-Relationship diagrams (ERDs).<br>ERDs in Enterprise Architect assist you in building conceptual data models through to generating Data Definition Language (DDL) for the target DBMS. |
| Eriksson-Penker Extensions | Eriksson-Penker extensions provide a framework for UML business processing model extensions, to which an Enterprise Architect can add stereotypes and properties appropriate to their business.<br>In Enterprise Architect, the Eriksson-Penker profile provides, through a set of stereotypes, a unique and powerful means of visualizing and communicating business processes and the necessary flow of information within an organization. |
| Gang of Four Patterns | Gang of Four (GoF) Patterns are 23 classic software Design Patterns providing recurring solutions to common problems in software design.<br>Enterprise Architect provides each Pattern through an icon in the Diagram Toolbox. |
| ICONIX | The ICONIX Process is a streamlined approach to Use Case driven UML modeling |

| | |
|---|---|
| | that uses a core subset of UML diagrams and techniques to provide thorough support of object-oriented analysis and design.<br>Its main activity is robustness analysis, a method for bridging the gap between analysis and design. |
| Mind Mapping | A Mind Map is an image-centered diagram used to represent semantic or other connections between words, ideas, tasks or other items arranged radially around a central key word or idea.<br>A Mind Map is used to generate, visualize, structure and classify ideas, and as an aid in study, organization, problem solving, decision making, and writing. |
| SoaML | Service Oriented Architecture (SOA) is an architectural paradigm for defining how people, organizations and systems provide and use services to achieve results. |
| SOMF | The service-oriented modeling framework (SOMF) is a service-oriented development life cycle methodology, offering a number of modeling practices and disciplines that contribute to a successful service-oriented life cycle management and modeling. |
| Extended Diagrams | Enterprise Architect provides an additional set of diagram types that extend the core UML diagrams for domain-specific models.<br>Also, the specialized modeling tools listed in the first part of this table each have their own specialized diagrams. |
| Inbuilt and Extension Stereotypes | Behavioral and Structural elements can be extended through the use of stereotypes; Enterprise Architect provides a number of inbuilt extensions. |
| Build Your Own Modeling Language | Enterprise Architect enables you to extend the scope both of your modeling and of the UML components you use, through the use of stereotypes, Profiles and Patterns to develop your own modeling applications. |

# Domain Based Diagrams

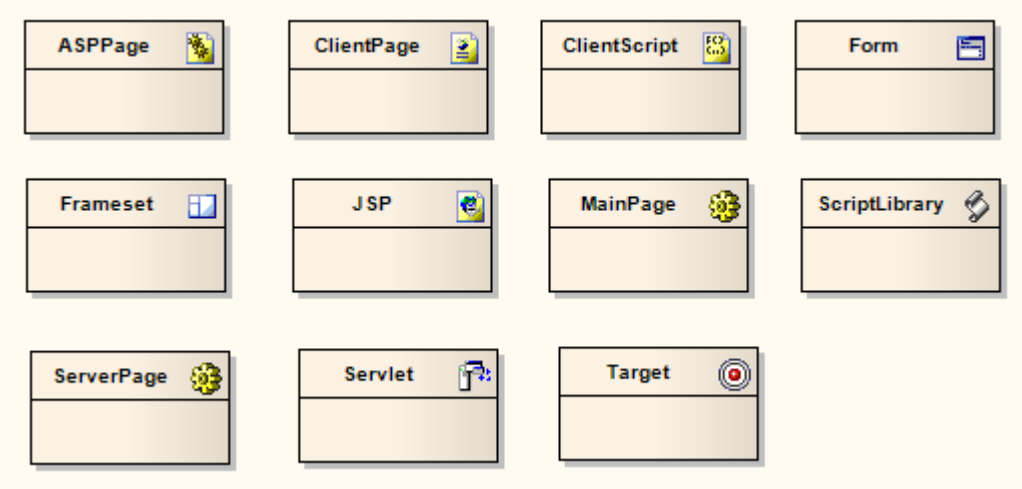
Enterprise Architect supports a wide range of modeling languages, such as UML, SysML and BPMN, but in addition to the diagrams that are defined as part of these languages Enterprise Architect has a rich set of additional (extended) diagrams, allowing you to create diagrams including Mind Maps, User Interface diagrams and Data Modeling diagrams; there is even a general purpose Custom diagram. This allows other disciplines such as strategic thinkers, user experience designers and scientists to contribute to the models and to create a repository of articulated knowledge that has not been possible before.

## Domain Based Diagram Types

| Diagram Type | Detail |
|---|---|
| Analysis Diagram | An Analysis diagram is a simplified Activity diagram, which is used to capture high level business processes and early models of system behavior and elements. |
| Custom Diagram | A Custom diagram is an extended Class diagram that is used to capture requirements, user interfaces or custom-design models. |
| Requirements Diagram | A Requirements diagram is a Custom diagram used to describe a system's requirements or features as a visual model. |
| Maintenance Diagram | A Maintenance diagram is a Custom diagram used to describe change requests and issue items within a system model. |
| User Interface Diagram | User Interface diagrams are Custom diagrams used to visually mock-up a system's user interface using forms, controls and labels. |
| Data Modeling Diagram | A Data Modeling diagram is a Class diagram used for representing database schemas. |
| Documentation | Virtual documents enable you to structure and filter your document and web reports by selecting, grouping and ordering individual Packages independent of the organization of the Browser window. |
| Business Modeling and Business Interaction | Business Modeling diagrams and Business Interaction diagrams enable you to model both the structure and behavior of a business system.<br>Business Modeling diagrams are based on a Class (UML Structural) diagram, whilst Business Interaction diagrams are based on a Sequence (UML Behavioral) diagram. |

# Web Stereotypes

Enterprise Architect supports a number of stereotypes for web page modeling, the graphical elements for which display with a graphical icon instead of the usual «stereotype» format. These stereotypes are only supported for Class elements. These are the various graphical icons and their associated stereotypes:




A similar set of web modeling elements and their relationships are also available through dedicated Web Modeling pages in the Diagram Toolbox.

## Set a web icon

| Step | Action |
|---|---|
| 1 | Create a new Class element in a diagram. |
| 2 | Display the Class 'Properties' dialog. |
| 3 | In the 'Stereotype' field, either type in the required stereotype name or click on the drop-down arrow and select the required stereotype (as named previously). |
| 4 | Click on the OK button.<br>The Class displays as in one of the examples shown. |

# User Interface Diagrams

User experience and user interface design have traditionally been modeled in a variety of tools that are separate from other disciplines, leading to a disconnect between these models and the other analysis and technology models.

Enterprise Architect allows you to model a wide range of user interfaces and platforms, including client software, web sites and pages, and mobile devices such as phones and tablets. It uses compelling representations of the physical devices and the platforms to make these models appealing and useful for walks-through with users. The elements in these models can also be traced to other elements in the repository, including design principles, requirements, use cases and user stories, stakeholders' concerns, information models, architecture and design models. StateMachine diagrams can also be created to represent the important states of the user interface, and these can be traced to testing models.

The User Interface diagram is an extended diagram type that provides a set of wire framing toolboxes with a rich palette of user interface elements for Android and Apple devices, as well as for web pages and dialogs. There is also a facility for modeling Win32® user interfaces, with a toolbox containing a wide range of controls such as Check Boxes, Spin Controls, Tree Controls and many more.

## Example Diagram

Example User Interface Diagram

## User Interface Diagram Element Toolbox Icons

| Icon | Description |
|---|---|
| Package | Packages are used to organize your project contents, but when added onto a diagram they can be use for structural or relational depictions. |
| Screen | A Screen element is used to prototype a User Interface screen flow. |
| UI Control | A UI Control element represents a user interface control element (such as an edit box). |
| Object | An Object is a particular instance of a Class at run time. |

## User Interface Diagram Connector Toolbox Icons

| Icon | Description |
|---|---|
| Associate | An Association implies that two model elements have a relationship, usually implemented as an instance variable in one or both Classes. |
| Aggregate | An Aggregation connector is a type of association that shows that an element contains or is composed of other elements. |
| Generalize | A Generalization is used to indicate inheritance. |
| | |

| | |
|---|---|
|  | A Realizes connector represents that the source object implements or Realizes its destination object. |

## Notes

- Using stereotyped Classes, you can model the design of a web page user interface
- The Enterprise Architect Professional, Corporate and Suite editions also include the MDG Win32 UI Technology, with which you can design user interface components that render more precisely as Win32 ® User Interface elements

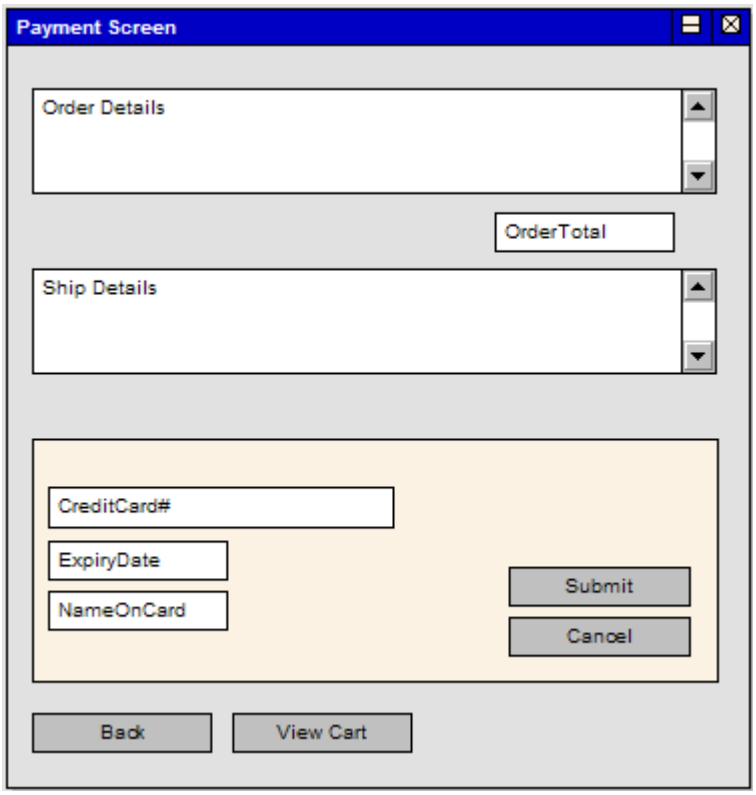
# Screen

A Screen is used to prototype User Interface screen flow. By using UML features such as Requirements, constraints and scenarios against User Interface diagram elements, you can build up a solid and detailed understanding of user interface behavior without having to use code. This provides an excellent means of establishing the precise behavior of the system from a user's perspective and, in conjunction with the Use Case model, defines exactly how a user gets work done.

Web pages can also be prototyped and specified rigorously using Enterprise Architect's custom interface extensions.

## Example

This example diagram illustrates some features of Enterprise Architect's screen modeling extensions that support web page prototyping. By adding requirements, rules, scenarios and notes to each element, a detailed model is built up of the form or web page, without having to resort to GUI builders or HTML.

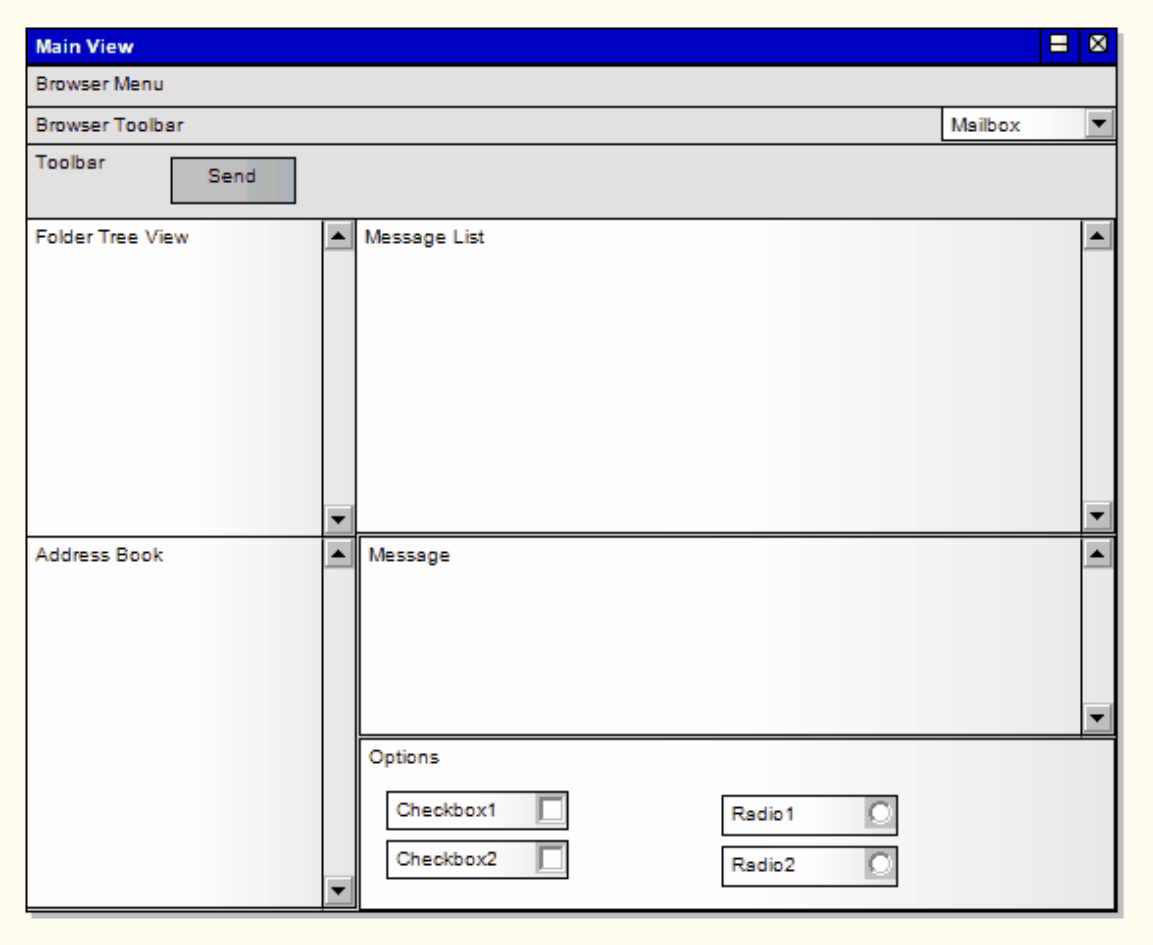



Enterprise Architect displays UI Controls as a range of special icons, depending on the stereotype used; for example, a Control stereotyped as a «list» displays with a vertical scroll bar.

## Toolbox icon

# Example User Interface Diagram

In this example User Interface diagram, forms, controls and labels are arranged on the diagram to define the appearance of a user interface screen and controls. UI Control elements can also be traced to other model elements linking the UI with the underlying implementation.




## Notes

- The Screen element is the parent of all the UI Control elements it contains; in the Browser window, expand the Screen element to list its child UI elements
- If you are designing more than one screen, and you want to move a UI Control element from one screen to another, you can do this in the Browser window - click on the UI Control element and drag it underneath the target Screen element; on the User Interface diagram, the UI Control element is removed from the source Screen and displayed in the target Screen

# UI Control Elements

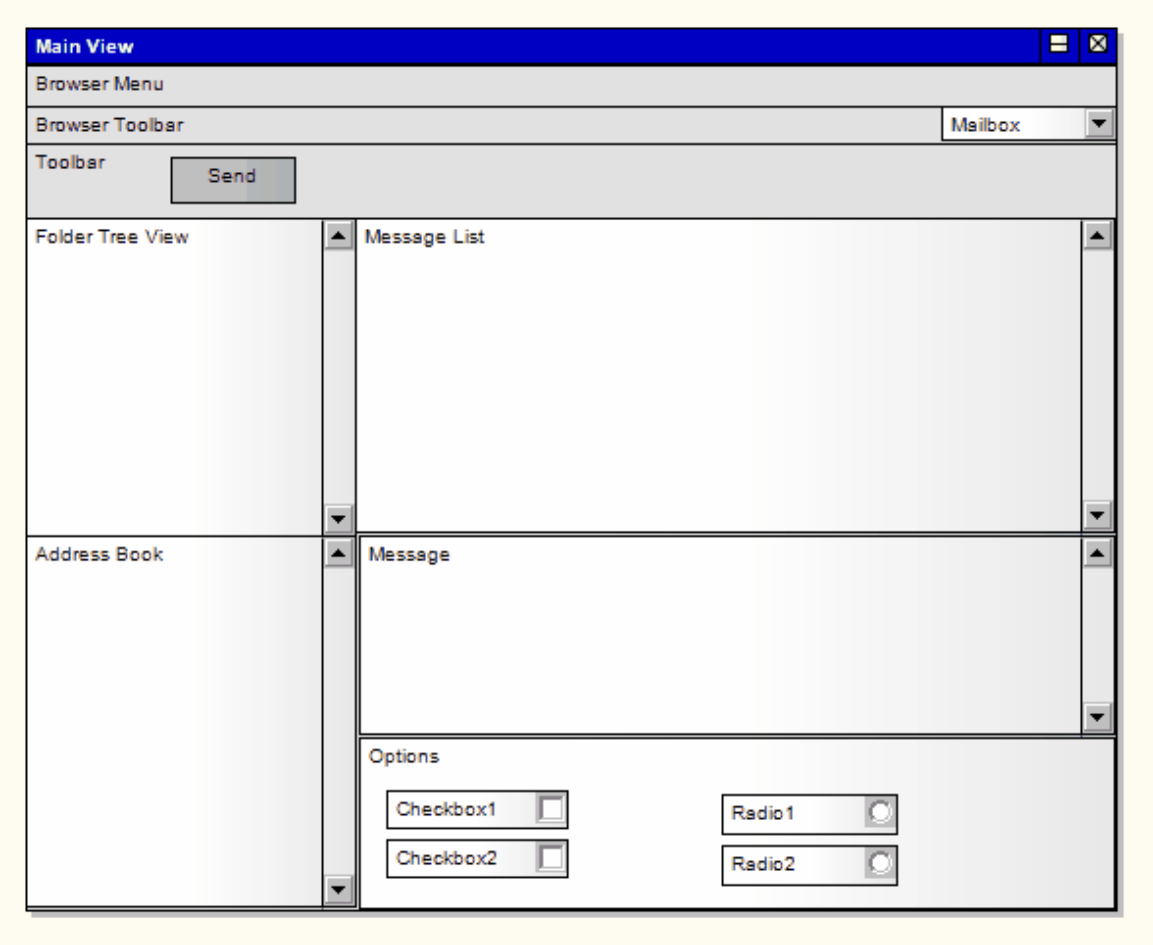
A UI Control element represents a user interface control element (such as an edit box). It is used for capturing the components of a screen layout and requirements in a Custom or User Interface diagram.

There are a number of UI Control elements available in the 'User Interface' page of the Toolbox. These include:

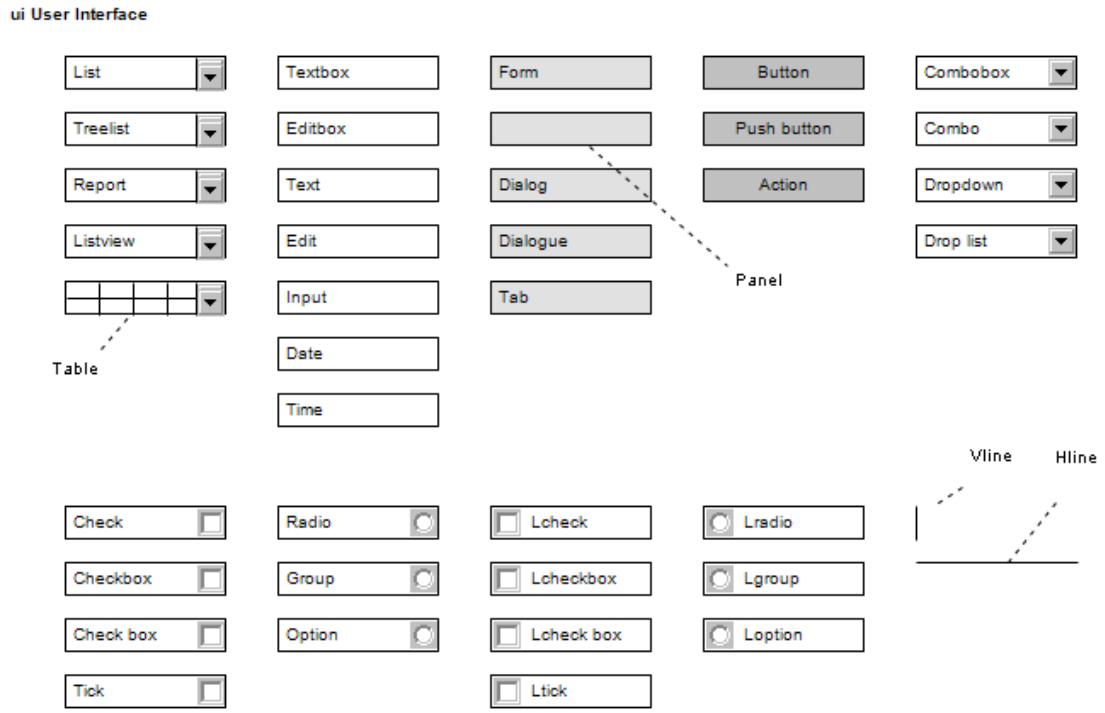
- List
- Table
- Text Box
- Label
- Form
- Panel
- Button
- Combobox
- Checkbox
- Checkbox (left hand side)
- Radio button
- Radio button (left hand side)
- Vertical Line
- Horizontal Line

## Example

The icons can be combined on a 'Screen' icon to represent the appearance of a user interface screen, as shown:




You can also extend the available icons by selecting other stereotypes in the 'UI Control Element Properties' dialog. The full set of available stereotypes is shown here; type or select the text in the 'Stereotype' field to create the corresponding icon.



## Toolbox icon

(where UI Control is the name of the user interface element type)

# MDG Technology for IFML

The Interaction Flow Modeling Language (IFML) provides system architects, software engineers, and software developers with tools that support the platform independent description of graphical user interfaces for applications accessed or deployed on such systems as desktop computers, laptop computers, PDAs, mobile phones, and tablets. The language was developed by the Object Management Group; the IFML specification (version 1.0. February 2015) is available from the OMG website.

The MDG Technology for IFML provides access to the OMG's UML Profile for IFML within Enterprise Architect.

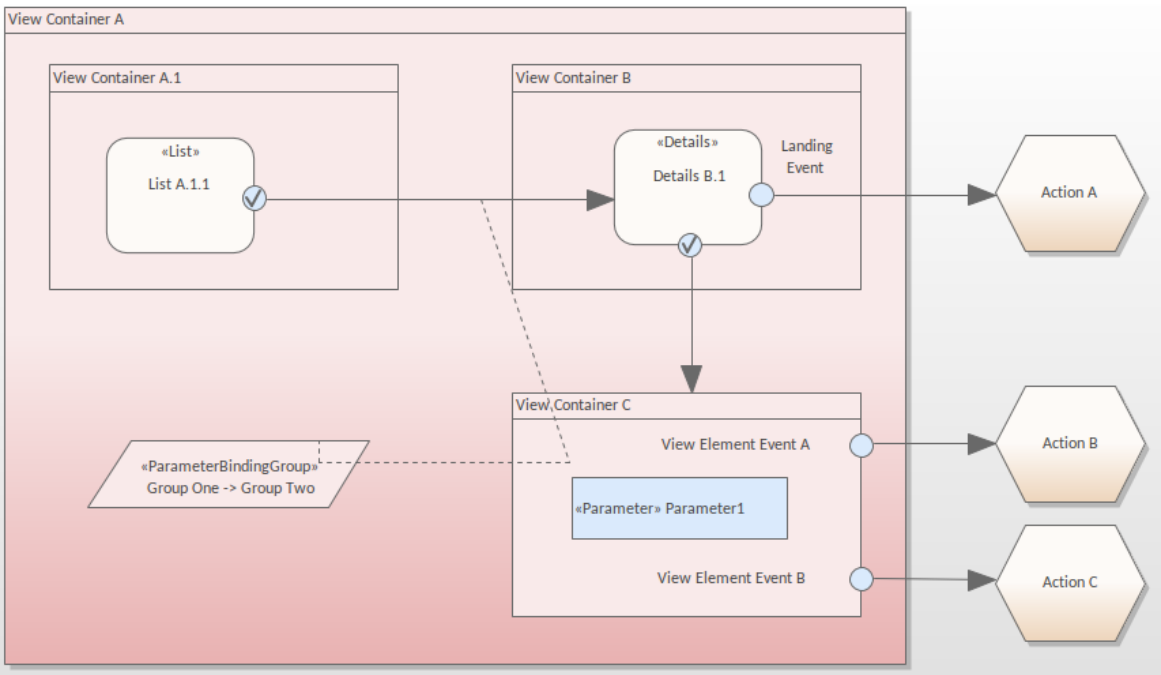
## IFML in Enterprise Architect

In Enterprise Architect you can model application interaction flows quickly and simply through use of the MDG Technology integrated with the Enterprise Architect installer. The IFML facilities are provided in the form of:

- Eleven IFML model patterns for Information Entry, Interaction and Wireframe, Searches and Desktop Applications, available through the 'Interaction Flow IFML' page of the 'Model Patterns' tab, in the Model Wizard
- Two IFML diagram types - IFML diagram and IFML Domain Model diagram - accessed through the 'New Diagram' dialog
- IFML 'Essential Concepts', 'Core' and 'Extensions' pages in the Diagram Toolbox

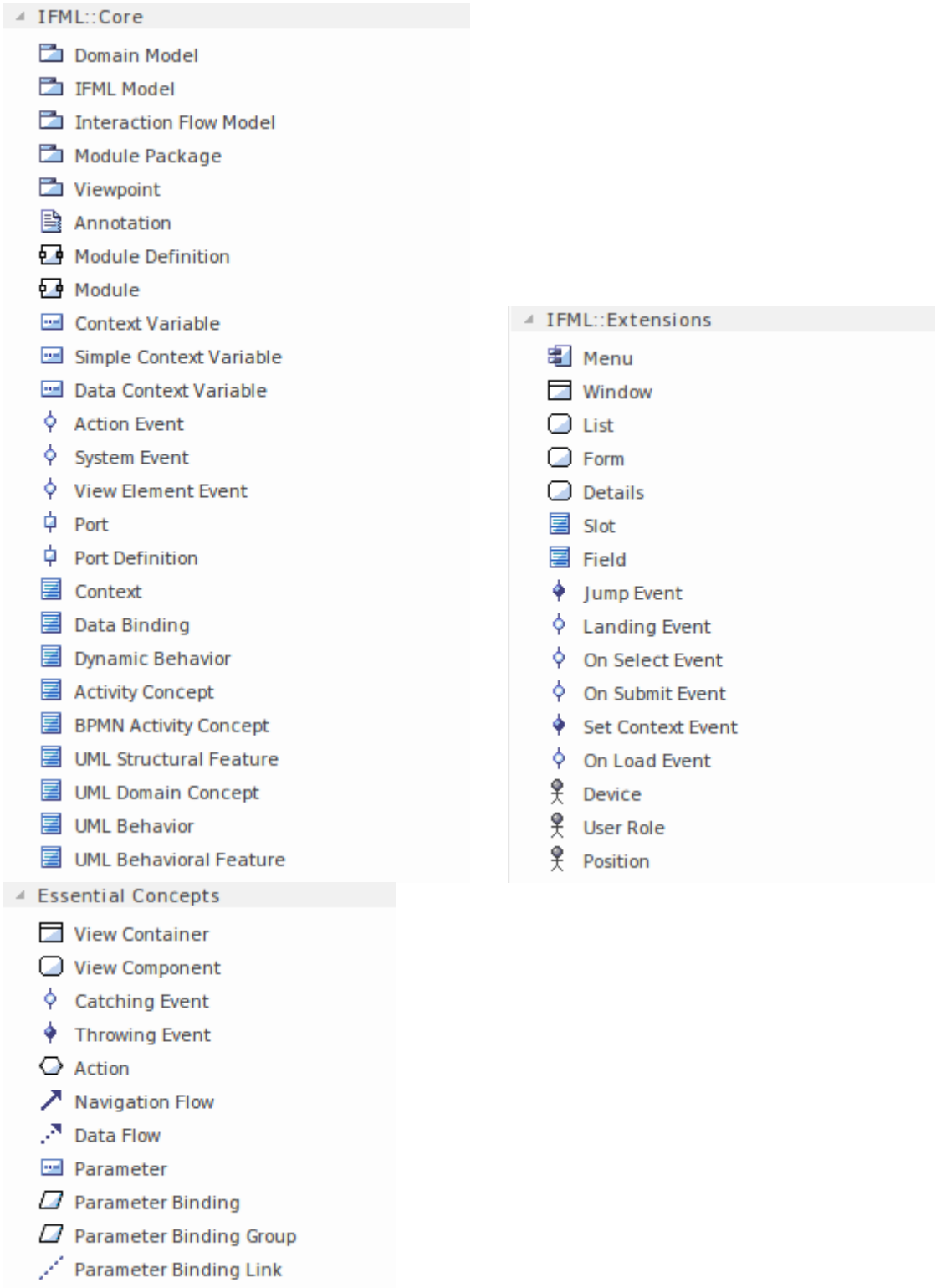
You can, if you wish, make the IFML Technology your default by selecting it in the MDG Technologies window and clicking on the Set Active button.

## Example Diagram




## IFML Toolbox Pages

The objects defined by the IFML Specification can be created in your model using the icons from these pages of the Diagram Toolbox:




Some objects from the IFML Specification are configured by Tagged Values on the main element type. For example, the View Container element can be set as a Default, Landmark or XOR View Container by setting the appropriate Tagged Value on the element to 'True'.