# Enterprise Architect

# Other Domains

Domain Modeling with Sparx Systems Enterprise Architect

Author: Sparx Systems

Date: 16/10/2024

Version: 17.0
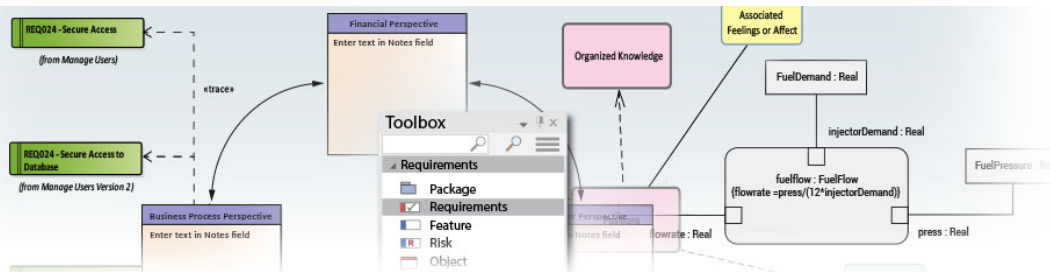
CREATED WITH ENTERPRISE ARCHITECT

# Table of Contents

# More Domain Models



Enterprise Architect is a multi-featured and comprehensive analysis and design, execution and testing tool that helps you to create models of enterprise, business, engineering, real-time and software systems. Its feature set is so rich that no matter what you want to do there is almost always a tool feature that will help you achieve the task. Enterprise Architect supports a wide range of frameworks, modeling languages and technology platforms and languages. Every discipline will benefit from using the tool, and when an entire team is using the tool the models created by each group can be seamlessly integrated into a whole that will bring great clarity and efficiency to the project or program of work and substantially reduce the risk of failure.

High level managers can create Mind Maps, and strategic thinkers can create stakeholder profiles and define business goals and drivers; Enterprise Architects can create road maps and application inventories, Analysts can describe current and future process models, and Requirements Analysts can create requirement models. On the solution side of the model, a Solutions Architect can detail the interfaces between applications, developers can create code stubs, and Testers can define test cases, just to detail a few

possibilities.

Enterprise Architect allows you to follow any project methodology, whether it is an Agile process or a formal military process or one of the standard frameworks such as TOGAF or Zachman. You are also free to set up your own modeling process, and Enterprise Architect has built in features for defining software development processes. Modeling can start at any point and in real projects it typically doesn't follow the sequences described in text books; Enterprise Architect allows you to work in a flexible way but supports you with a versatile set of tools.

Enterprise Architect has a flexible feature that helps you to create Profiles of the UML so even if you cannot find a suitable modeling solution built into the core product, you can extend the tool by creating your own profile.

# 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, including Mind Maps, User Interface diagrams and Data Modeling diagrams; there is even a general purpose Custom diagram. This allows several specialists 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 | Business Modeling diagrams and Business Interaction diagrams enable you to model both the structure and behavior |

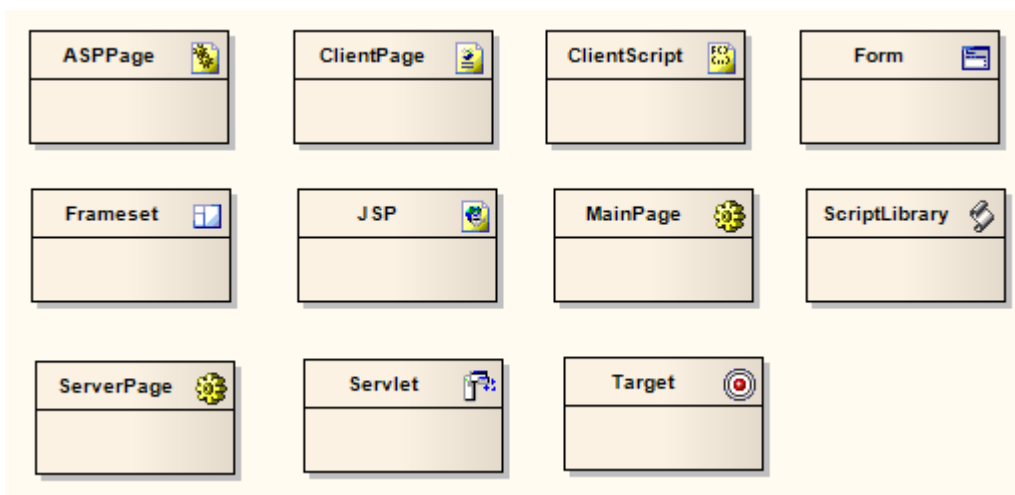| Interaction | of a business system. |
|---|---|
| | Business Modeling diagrams are based on a Class (UML Structural) diagram, whilst Business Interaction diagrams are based on a Sequence (UML Behavioral) diagram. |

# Modeling Disciplines

**Modelers from all Disciplines and Domains can Create Industry Standard Models**

Regardless of the discipline or your specific field in the system development life-cycle you will find modeling tools and options to create expressive and industry standard models. Regardless of whether you are creating new system components or selecting a commercially available product, or if you work in a strategic, business, technology or engineering division of an organization Enterprise Architect allows you to  create collaborative models that show how all facets of the product fit together. The collaboration features ensure that requirements, design and implementation choices are all resolved as a solution is developed.

# 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.
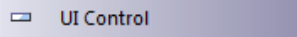
Enterprise Architect allows you to model a wide range of user interfaces and platforms, including client software, web sites and pages, and mobile devices such as phones and tablets. It uses compelling representations of the physical devices and the platforms to make these models appealing and useful 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 Toolbox pages 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 Toolbox pages containing a wide range of controls such as Check Boxes, Spin Controls, Tree Controls and many more.
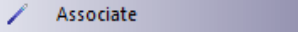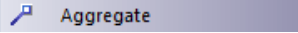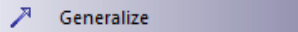
## Example Diagram

[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 used to depict the structure and relationships of your model. |
| 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. |
| Realize | 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, unified and Ultimate editions also include the MDG Win32 UI Technology, with which you can design user interface components that render more precisely as Win32 ® User
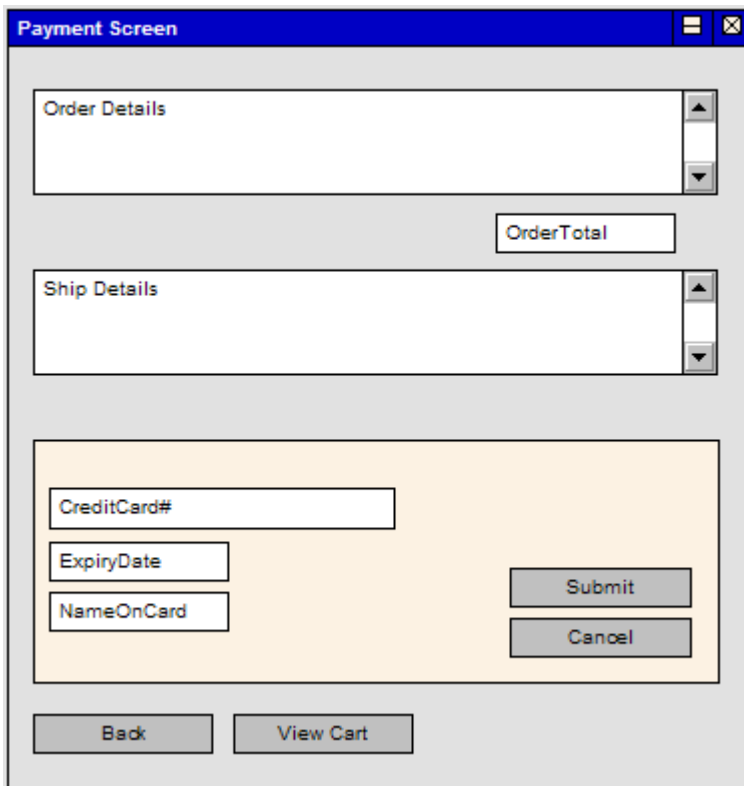
# Interface elements

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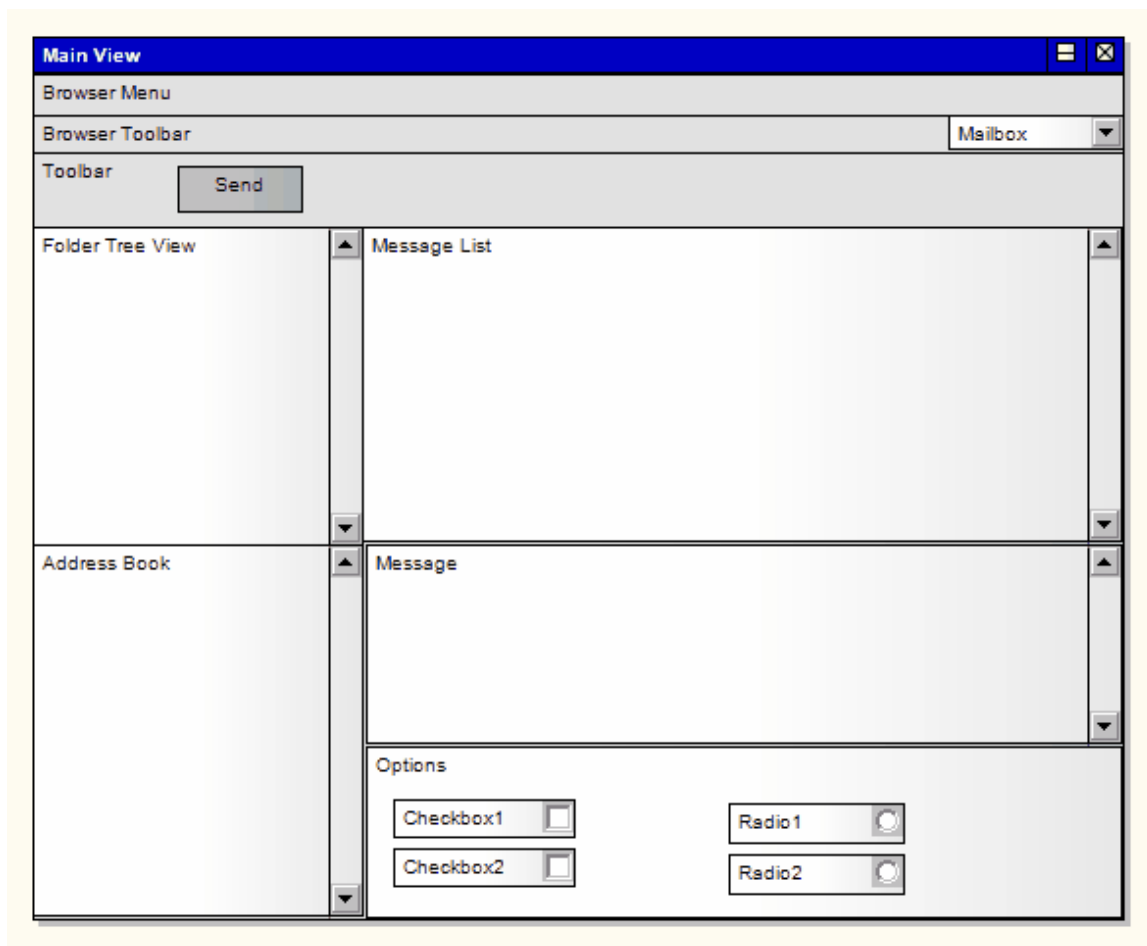
Enterprise Architect displays UI Controls as a range of special icons, depending on the stereotype used; for example, a Control stereotyped as a «list» displays with a vertical scroll bar.

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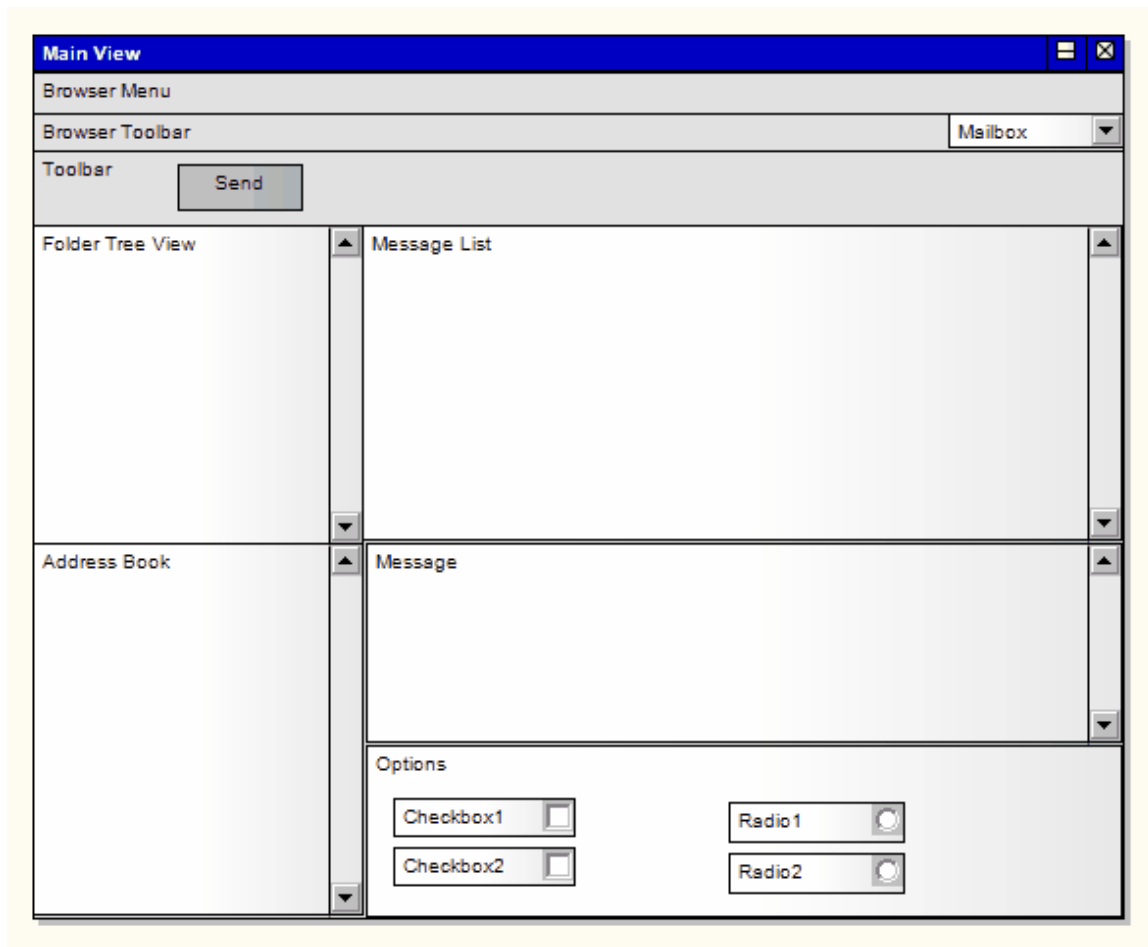
There are a number of UI Control elements available in the 'User Interface' page of the Toolbox. These include:

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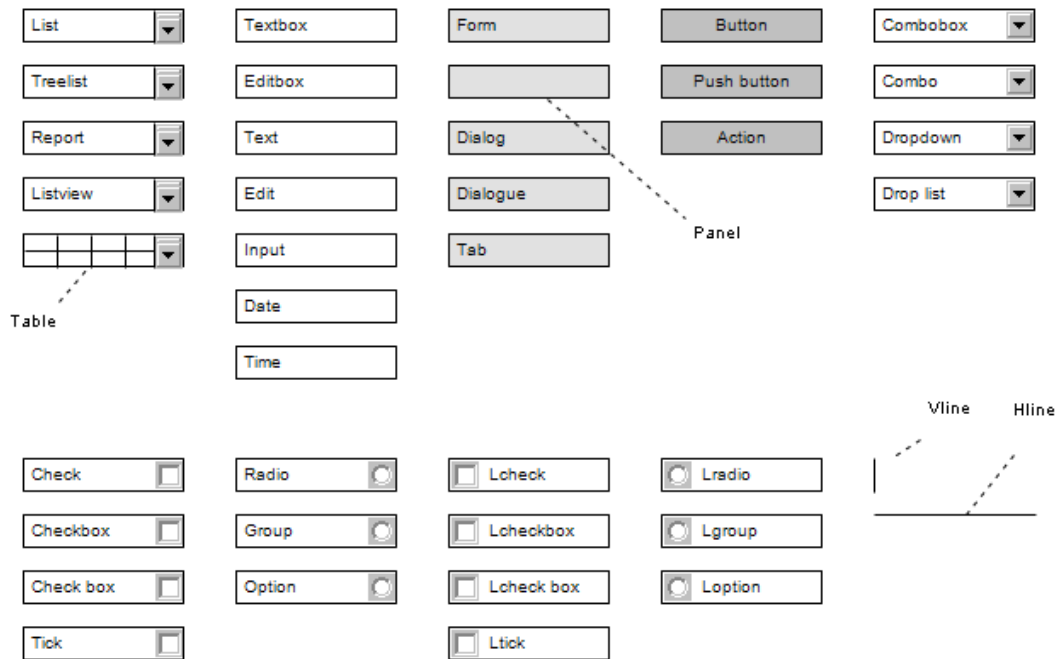- Horizontal Line

## 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.

**ui User Interface**

| | | | | |
|---|---|---|---|---|
| List | Textbox | Form | Button | Combobox |
| Treelist | Editbox | | Push button | Combo |
| Report | Text | Dialog | Action | Dropdown |
| Listview | Edit | Dialogue | | Drop list |
| *(table)* | Input | Tab | | |
| | Date | | Panel | |
| | Time | | | |

Table

| | | | | |
|---|---|---|---|---|
| Check | Radio | Lcheck | Lradio | Vline   Hline |
| Checkbox | Group | Lcheckbox | Lgroup | |
| Check box | Option | Lcheck box | Loption | |
| Tick | | Ltick | | |

# Toolbox icon

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