

Unit 12

Tersus Platform

Structure:

- 12.1 Introduction
 - Objectives
- 12.2 Introduction to Tersus Platform
- 12.3 Features of Tersus Platform
- 12.4 Exploring Tersus Platform
 - Tersus Studio
 - Tersus model libraries
 - Tersus server
- 12.5 Modeling the Basic Display
 - Modeling the logic behind the screen
- 12.6 Summary
- 12.7 Terminal Questions
- 12.8 Answers

12.1 Introduction

In this unit we are going to discuss one of the Visual application development platform called Tersus. Tersus is a dual licensed open source and proprietary visual programming platform for software development. It supports in application development especially for web based application. In this platform rather writing a code you can use the drawing tool to develop the business logic. The Tersus Modeling language supports in designing user interface, both server and client side processing. The Tersus shares many features of data flow programming language.

Tersus can be used for Web application development; here the server side programming can be designed with the help of AJAX techniques. Tersus modeled application execution is taken care by the Tersus server.

Like visual studio, Tersus also has the integrated development environment it is an extension of Eclipse platform. Tersus IDE helps the programmers to design the application in a graphical environment. Tersus has the debugging facility that captures every transaction that the application executes. This helps in tracking the program and flow of data in each transaction. Here we are going to explore the components and architecture of Tersus visual

platform. We will also discuss, one small application to understand the model design and the logic behind the screen.

Objectives:

After studying this unit, you will be able to:

- explain the Tersus visual platform
- discuss the features of Tersus platform
- elaborate the architecture of Tersus platform
- list and explain the components of Tersus platform
- develop and execute simple application in Tersus platform

12.2 Introduction to Tersus Platform

The Tersus technology is used to create various software applications, from small strategic applications to high end, operation critical systems for processing financial transactions. Tersus is specifically suitable for complex applications accumulated in the combination of built-in components; user developed components and Web Services.

Following are three components of Tersus platform

- **Tersus Studio:** Tersus studio is considered as extended form of Eclipse platform, helpful for the developers and experts to describe the functions of applications in a graphical way The application environment used by modellers (programmers and business professionals) to visually define application capabilities. In this way, Tersus Studio contributes to the Eclipse ecosystem;
- **Tersus Model Libraries:** consist of built-in blocks to build software applications.
- **Tersus Server:** This implements the modeled solutions and update the database based on the demand.

The Tersus IDE supports three types of application operations they are

- Create
- Deploy
- Maintenance

Tersus Framework:

The Tersus platform is a visual programming framework that allows you to create software applications without having to know how to code. Tersus is a dual licensed open

source and proprietary visual programming platform for software development. It supports in application development especially for web based application. In this platform rather writing a code you can use the drawing tool to develop the business logic. The Tersus Modeling language supports in designing user interface, both server and client side processing. The Tersus shares many features of data flow programming language.

The different features of the Tersus framework.

1. The Tersus technology is used to create various software applications, from small strategic applications to high end
2. It servers as operation critical systems for processing financial transactions.
3. Tersus is specifically suitable for complex applications accumulated in the combination of built-in components;
4. **Language independence:** Tersus is incredibly flexible, it allows you to build your pages exactly how you want by using the powerful short code generator and page options, model names and GUI can be in any language. Also the Tersus platform supports in multiple languages.
5. **Built-in templates and prototypes:** Tersus platform supports with lots of model templates and the model prototypes. It has the ability to reuse any model in multiples context thus reduced the recoding.
6. **Importing Services:** Importing WSDL (Web Service Definition Language) definitions of web services as Tersus building blocks

Components of Tersus framework.

The two major components to be considered for framework are modelling language and second one is server .

Let us understand in detail now it helps in language modelling , were Tersus is incredibly flexible, it allows you to build your pages exactly how you want by using the powerful short code generator and page options, model names and GUI can be in any language. Also the Tersus platform supports in multiple languages. It supports Built-in templates and prototypes: Tersus platform supports with lots of model templates and the model prototypes. It has the ability to reuse any model in multiples context thus reduced the recoding.

Importing Services: Importing WSDL (Web Service Definition Language) definitions of web services as Tersus building blocks

Other important component is server , we will see it in detail further.

1. Modelling Language: The first component is considered to be modelling language, it basically specifies the fundamentals requirements of the entire web applications, it eliminates the syntactical errors that is generated during web application development process , and it also allows you to specify the model that is been used to develop the web application. Tersus platform supports with lots of model templates and the model prototypes. It has the ability to reuse any model in multiples context thus reduced the recoding. Importing WSDL (Web Service Definition Language) definitions of web services as Tersus building blocks
2. Component Server: Server is the second component of the Tersus platform

it supports in code free application development. It is possible to create and run simple web application in a minute. External web services can be used for designing building blocks. Model exchange is possible with sharing libraries modules and applications.

Building Block of Tresus Frame work:

The building blocks of the Tersus platform have been illustrated here.

1. Firstly we have considered Data types: Tersus common library has all the basic data types. You can use these data types for models also it allows constructing composite data types. This is useful to restrict the allowed content for data elements (valid values). Also a special data type may be defined specifying the constraints on its valid values.
2. Second one is Application modules: it supports in code free application modules. It is possible to create and run simple web application in a minute. External web services can be used for designing building blocks. Application Model exchange is possible with sharing libraries modules and applications.
3. Third one is document modelling actions which helps us in handling the documents that describes the actions to be performed .
4. Next building block is mathematical function that involve various built-in mathematical libraries were in mathematical functions can be directly used and accessed .
5. The fifth building block is GUI elements that supports Look and feel customization through [CSS](#). The look of Tersus applications are controlled through HTML, CSS and

JavaScript.

6. And the last building block is database action which deals in handling database.

Data types and GUI elements are two building blocks of the Tersus platform which is considered to be special and important building components of Tersus, because the data types The type of data that is used in the web application is defined by data types, Tersus common library has all the basic data types. You can use these data types for models also it allows constructing composite data types. This is useful to restrict the allowed content for data elements (valid values). Also a special data type may be defined specifying the constraints on its valid values.

Document Handling Actions:

The document handling actions with examples have been illustrated here. Some actions include paper management as well as programme execution for example, let us consider Sending e-mail addresses, reading an Excel spreadsheet and creating a PDF document, among other things

When an application is executed, important events and errors are reported to a log file. It is possible to record any step of the execution and later trace the details of execution (useful while developing a new solution or for debugging).

- Tracing is displayed graphically on the model diagrams, showing both process flow and data values.
- The user can view the tracing at high-level diagrams only or drill down to any level of the model.
- Tracing can be played forwards (to see the natural execution flow) or backwards (for root cause analysis).

Mathematical functions with examples have been explained here. Tersus platform supports with lots of mathematical model templates and the model prototypes. It has the ability to solve mathematical problems, model in multiples context thus reduced the recoding. Mathematical functions are used to perform operations on operands many various mathematical operations can be executed using the tersus mathematical model.

The last two components of the Tersus framework have been explained here. When it comes to computer programming, an Application Module is a logical container for coordinated items linked to a specific job. By providing a single connection per Application Module at runtime, Application Modules

simplify the process of defining and executing transactions. Application modules which are basically used for a variety of purposes, including authorization, user interface. In particular, it contains the data model for a given task as well as any necessary custom code. Updates to client data, new order creation, and wage rise processing are all examples of application tasks. Then the last components is Data base Action which describes the different operations that can be carried out on the database , some of the database operations like insertion ,deletion, updation and so on.

The steps involved in the creation of web applications using Tersus have been shown here.

- Tersus helps in developing different kinds of web and mobile applications.
- Tersus used in developing web applications like sales dashboard, music library, games, etc.

First steps is like create a diagram, then automate the development of the server side processes then application becomes live, and automatically it creates a screen layout, and finally the process defines client side behavior.

Working of tersus framework:

- Tersus have been appointed for various frameworks like Fusion 21, reallices, Eastern procurement, consortium procurement, etc.
- Tersus provide a palate which have buttons, labels, etc.
- The very first step is to provide a screen layout , where we can use buttons, tables, and many more controls to provide the look and feel of the user interface , second we can also add the fields by validating it , further to proceed the second phase is to process on the client and server sides which are generated automatically based on the functionality write to be executed . So this client server sides can be generated and used by the visual debugger to visually monitor the model.

The advantages of the Tersus framework:

- As we have seen so many features , Tersus is a visual programming platform that eliminates the need to write code in favour of a graphical interface for developing enterprise-level web applications.
- When developing large-scale online applications, it is said to be language independent because the model names and user interface can be written in any language.
- Model templates and prototypes are available so that we can use the templates to build the protocols, the system also includes a visual debugger which helps in

detecting the errors very easily, one more feature of tersus is Instead of writing the code.

- The web application developer must only define the model, making web application development simple.

Process involved in creation of application platform:

Tersus platform and different steps required to make a web application in the Tersus platform. First and foremost tarsus platform is used to create a web application, after creating the web application , we can deploy the web application and maintain the existing web application Using Tersus studio and Tersus modules, users may easily and quickly design responsive user interfaces. Tersus unifies the development of user interfaces, client-side behaviour, and server-side processing into a unified visual language. Tersus uses AJAX "under the hood" to build online applications with rich, responsive user interfaces.

1. Web application creation
2. Web application deployment
3. Maintenance of Existing web application

1. **Web Application Creation:** Started by defining a Model Hierarchy, the developer starts at a top-level diagram representing the whole system, and then continues with an iterative top-down refinement process, drilling down from each model to specify its components. This process helps to represent graphically the whole model hierarchy. Using this model it is possible for the developer to fully and accurately represent the required business logic in a visual and interesting way. A developer can create a visually appealing representation of the necessary business logic, which is both complete and correct.

2. **Web Application Deployment:** Once the models are ready, will be saved as a hierarchy of XML files that are read by the Tersus Runtime Engine. Now the role of engine is to execute the operations that are defined by the models at various levels like user interface, database and server-side processing operations. This platform supports in capturing the execution process that may be helpful for future or for any cross reference in case of error handling

3. **Web Application Maintenance:** An existing application is done by revising its model changing the business flow, adding new components, or

deactivating duplicated components. Towards the end of the completion of the required modifications, the same application can be redeployed immediately.

Example:

Sample web application creation is explained in this slide, initially open the tersus studio and create a new application project , then we will start modelling , modelling can be done by creating a form for entering a requisition, then open the application root model in the model editor ,

- To create the View called Open Request: In the Palette , make sure that the Display category is open. Later Click on the View template to select it and name it as open requisition
- Add a Button (New Request)
- Select the button template from the Display category of the palette and insert it into the Open Requisitions view. Name the button as New Request.
- Click on the launch application to view the application.

Example 2:

- Open tersus studio
- Create a Popup (Enter New Request)
- Model the popup form that appears when the button is clicked: Double-click the New Request button to zoom into it, then select the Display/Popup template from the palette to insert it into the New Request element. Later, Name it Enter New Request.
- Add Display Elements to the Popup
 - First, create a label ("Description:"), which prompts the user to describe the requisition:
 - Double-click the Enter New Request popup to zoom into it. Select Display/Label template and drop it into the upper part of the popup. Name it as 'Description:'.
 - Select the Display/Text Area template and drop it below label and name it as description.
- Rename the model:
- The Enter New Requisition popup contains the pre-built button, OK. Replace it with submit by zoon-in technique.

Importance of Tersus Framework Model:

Firstly the models are defined as meta data, then what is said to be meta data ?, the definition of meta data is referred to the data which holds the information of other data is

known as the meta data, however here the models are said to be as meta data because it just keeps the information of the data that has been accessed, these models are initialized when the programme is running, the second phase is called as model displays which will exhibit the functionalities of the program in running mode.

The semantic models of the Tersus platform have been defined here. The semantic data model is a method of structuring data in order to represent it in a specific logical way. It is a conceptual data model that includes semantic information that adds a basic meaning to the data and the relationships that lie between them. Multiple displays would be allowed in a display screen if semantic models are used. Button, view, table and other components are all used in display models. The semantic data model is an up-and-coming strategy built on semantic concepts that yield data with predetermined internal structure. When taken out of context, a single piece of information or word has little value to a human being.

Components of semantic model:

The components of the semantic models have been illustrated here. we have four components like

1. System
2. Process
3. Data structure
4. Data item.

Each component has a role to perform individually that together combines and gives the semantic data model features. Let us see the functionality of each and every component in detail

The first component is considered to be system, the root of the hierarchy is determined by a system component, Web application logic is typically represented by hierarchical models

The second component is process, where a process is the set of actions that makes up an execution where the environment is setup to take care of smooth execution of the program, the third component is data structure which deals in a data model, data types stand in for collections of information found in a database.

The last one is data item where it indicates the components of the data model known as

"data items" stand in for discrete chunks of information found in a database.

Components of Tersus Development platform:

Three components of the Tersus development platform have been shown here.

1. Tersus studio: Tersus is a visual programming platform for creating rich web and mobile applications.
2. Tersus server: The Tersus server executes the modelled applications.
3. Tersus model libraries: Tersus libraries consist of many templates and models that are useful for visual programming.

Architecture

The model of a typical application consists of Systems that is high level modules, Displays through GUI components, Processes means the activity units, and Data Structures and Data Items and information used by the application. Processes and in certain cases also systems and displays can receive and send out data through Slots. The flow of data between processes, as well as the sequencing of processes, is governed by flows. A flow appears in the model diagram as an arrow between two model elements, each of which is a slot or a data element. In fact, the Tersus modeling language is more than a visual language. Slots provide for the reception and transmission of data between processes and, in some situations, between systems and displays. The Tersus platform provides Executable Graphics, where the Tersus server is the execution engine that runs the modeled applications.

Tersus is used to create a range of software solutions, mainly web applications. Companies who choose the Tersus platform have done so due to its ease of use, increased productivity, and the high quality of the resulting applications. The usage of Tersus also widens the development teams, as it helps less skilled developers take part in the development process.

The platform includes:

Tersus Studio, the IDE used by modelers. It manages projects, each containing the models and resources of one application. The application models are saved as a set of XML files, each containing the details of all models in a certain package within the project.

Model Libraries, containing building blocks for assembling applications. Tersus Server, which executes the modeled applications and performs the required database updates. It contains an embedded application server (Tomcat) and an embedded database server (HSQLDB), which allow for immediate testing of the modeled applications. External application servers and database servers can be used to deploy applications operationally.

Self Assessment Questions

1. _____ is used by developers to define function using graphical way.
2. The movement between the process is ruled by the _____.
3. Tersus application are executed with the help of Tersus studio. State [True/False].

12.3 Features of Tersus Platform

Following are the features of Tersus platform

Language independence: Tersus is incredibly flexible, it allows you to build your pages exactly how you want by using the powerful short code generator and page options, model names and GUI can be in any language. Also the Tersus platform supports in multiple languages.

Built-in templates and prototypes: Tersus platform supports with lots of model templates and the model prototypes. It has the ability to reuse any model in multiples context thus reduced the recoding.

Importing Services: Importing WSDL (Web Service Definition Language) definitions of web services as Tersus building blocks

Data types: Tersus common library has all the basic data types. You can use these data types for models also it allows constructing composite data types. This is useful to restrict the allowed content for data elements (valid values). Also a special data type may be defined specifying the constraints on its valid values.

Look and Feel: Look and feel customization through CSS. The look of Tersus applications are controlled through HTML, CSS and JavaScript.

Testing and Debugging

- When an application is executed, important events and errors are reported to a log file.

- It is possible to record any step of the execution and later trace the details of execution (useful while developing a new solution or for debugging).
- Tracing is displayed graphically on the model diagrams, showing both process flow and data values.
- The user can view the tracing at high-level diagrams only or drill down to any level of the model.
- Tracing can be played forwards (to see the natural execution flow) or backwards (for root cause analysis).
- An application's execution generates a log file that records critical events and errors. Both the process flow and data values can be seen graphically on the model diagrams that depict the tracing.

Easy working environment: As discussed earlier it supports in code free application development. It is possible to create and run simple web application in a minute. External web services can be used for designing building blocks. Model exchange is possible with sharing libraries modules and applications.

Self Assessment Questions

4. WSDL stands for_____.
5. During the execution of application the errors are reported to _____.

12.4 Exploring Tersus Platform

To explore tersus platform we should have the knowledge about the three components that includes Tersus studio, Tersus model libraries and Tersus server.

12.4.1 Tersus studio

Installing Tersus studio using installation kit

1. Download the Windows installation kit executable, available online
2. Execute the file to launch the installer.
3. Follow the instructions given by the installer.

The kit includes everything needed to start modeling immediately (Embedded JRE, Eclipse Platform Runtime, Sample applications, Tutorial). The Windows installation kit may occasionally install an earlier version, of Tersus. However, when installation is complete, and the studio is launched, it will automatically check for updates and guide you through the required

download and installation process.

Tersus Studio is an app developed by a Israel-based company as a open-source platform. Tersus is fundamentally a GUI that empowers the user to create Web-apps using simple visual flowchart-based methodology, without ever touching one line of code. The program is based on Eclipse, an IDE developed for the OSS-community. So it runs almost like an SDK, and if you're comfortable with navigating around Eclipse, It will take some time to get familiar with Tersus. One more thing, Tersus is written with a Java backbone, so Web-apps created with Tersus are also Java-based.

Opening a new project in Tersus is similar to open a blank screen like any other simple editors. There you can view two distinct parts for Web-app creation; they are end-user interface and server-side actions. Creating an end user interfaces are very simple, just it is enough to drag textbox, table, grid icons or controls from a Photoshop-like palette on the right into the canvas. On the editor these controls can be labeled and given some kind of RDF (Resource Description Framework) relevance. Tersus, when it compiles, will automatically generate these as XHTML-compliant code that also looks very consistent. It is very simple and easy that the CSS will easily take care of the design of these fields, and they have some taxonomy and custom classes that CSS-users can easily style with.

But we all know creating the end-user interface is technically easier than the backend of it. Getting users to fill up a form is way easier than managing that information automatically, sending it to a database for storage and retrieving it easily and safely etc. Here comes the major role of Tersus platform. Getting data to store in a database is a simple matter of dragging a database block onto the canvas, creating fields in the new block, and then linking the text-fields to the database fields.

Many common SQL queries/actions are supported out of the box, all in the form of draggable icons. And if advanced SQL queries are needed, Tersus allows these to be coded manually and the same can be saved for future use. Beyond all this, the most impressive was the sense of scale achievable. Simple Web-apps could just be a mailing list form, but complex ones require many conditionals and many inputs to execute.

When new blocks are created in Tersus, it zooms into them, thus allowing each block to have its own complex hierarchy of data. Zooming out is as simple as clicking outside the block, and you are instantly back at the bigger

picture. This simplify may lead to Web-app production issues, because the reductionist methodology has proven to be very thorough and manageable. Figure 12.1 is depicting the screen shot of Tersus studio.

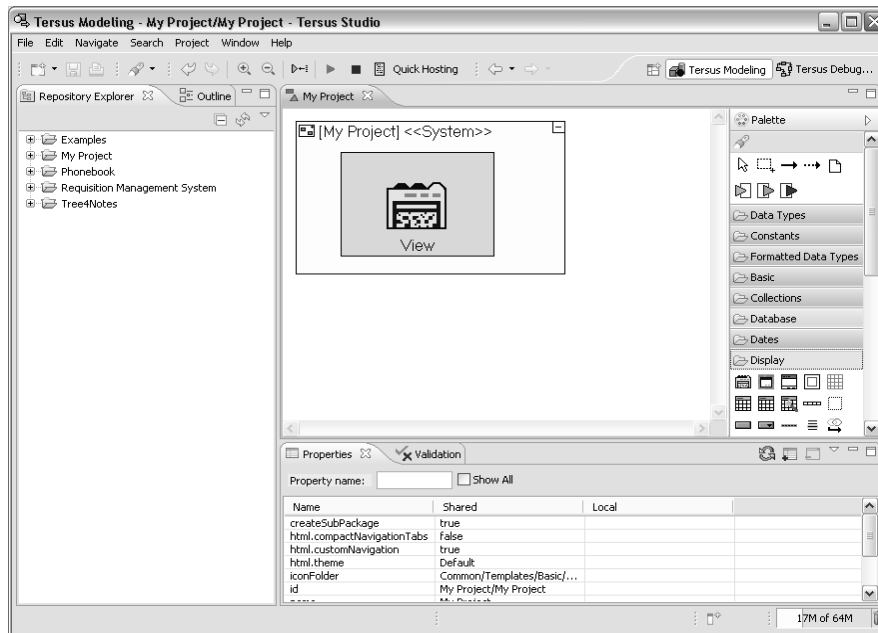


Fig. 12.1: Tersus studio

Tools in Tersus Studio

Tersus Studio, supports with multiple tools to support modeling, such as the Model Editor, Outline, Repository Explorer, Template Library, and embedded application and database servers.

a) Eclipse platform

The Tersus Studio uses the Eclipse platform, which is an industry standard IDE framework, providing various features and significant flexibility, through the menus and toolbar, including the possibility to rearrange the display to suit your taste.

The Eclipse platform uses the notion of a Perspective displaying one or more Views, arranged in a specific way. Switching between perspectives changes the make-up and arrangement of views. The figure 12.1 displays the Tersus Modeling perspective which, by default, includes the following views: Model Editor, Palette, Outline, Repository Explorer, Properties and Validation.

To switch to a different perspective, use Window -> Open Perspective

To change the arrangement of views in a perspective, click on a specific view's title bar, drag it around and drop it in its new position. If you need more information pertaining the features provided by Eclipse, can be referred from Eclipse platform help system (accessible through Help -> Help Contents).

Before we continue, make sure the Tersus Modeling perspective is displayed. To find out which perspective is currently in use, take a look at the Eclipse window title bar. If it does not start with Tersus Modeling, then another perspective is currently displayed. If this is the case:

Select Window -> Open Perspective -> Tersus Modeling

Or, if the Tersus Modeling option does not appear:

Select Window -> Open Perspective -> Other...

Eclipse should switch to the Tersus Modeling perspective, displaying the perspective name on Eclipse window title bar as shown in figure 12.2.

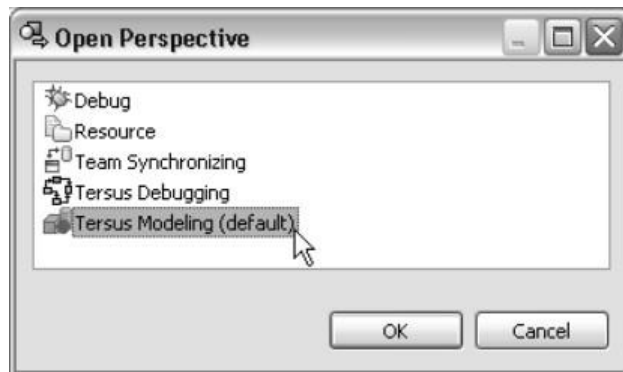
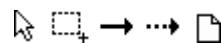


Fig. 12.2: Tersus modeling selection window

b) Palette in model editor


1. Editing Tools




Used to define and control the layout of elements in the editor.




(Select) Selects and moves elements in the editor.

 (**Marquee**) Selects all elements in the specified region from the editor.

 (**Flow**) Creates a (regular) flow between elements in the editor.


 (**Remove**) Creates a remove flow between elements in the editor.

 (**Note**) Adds a note (documentation/comment) to an existing element in the editor


2. Slots



Used to define input/output *ports* of processes.

 (**Trigger**-Green color) A port used to activate and pass input into a process

 (**Exit**-Grey color) A port used to pass output from a process

 (**Error Exit**-Red color) A port used to pass errors (exception) from a process

3. Template element

Templates are predefined elements which serve as building blocks for modeling. There are several categories of templates (such as Data Types, Database, Display etc.), and each category contains several templates as shown in figure 12.1 under the palette.

4. Search button ()

Provides the option to search for a specific template in the Palette by name or part of it.

c) Inserting a new element to the model

When you select a template from the Palette and return back to the editor. The mouse pointer changes to signify that a new element is going to be inserted into the model. There are 2 methods of insertion from the palette:

Click – Creates a new element with default size (hinted by a dashed rectangle).

Click & drag – Creates a new element with the size specified by the user.

New elements may also be inserted from the Repository and the Outline, using drag-and-drop. They will be created with a default size.


Selecting an Element

When you click on an element displayed in the editor, it becomes selected, marked with a frame, as in the following screenshot:



Fig. 12.3: Selected button control

Selecting multiple elements is possible using the following methods:

- After selecting the first element, hold down the [CTRL] key and continue to select additional elements.
- Use the Marquee tool () in the palette to specify a selection area in the editor. All elements in the marked area will be selected.
- When an element has been selected, drag it around to change its position in the editor.

Resizing an element

After an element has been selected it can be resized by dragging any of the 8 selection anchors which appear around the selection frame.



Certain types of elements (specifically Display, Process and System elements) resize differently when the corner anchors are used. These elements may contain other elements, therefore when the corner anchors are used to resize, the aspect ratio of the element and its sub-elements (width-to-height ratio) is maintained.

Drill-down

Models are hierarchical, meaning that an element may contain other elements (making up sub-models), and each of these sub-elements can in turn contain additional sub-models, and so on.

The model editor provides drill-down functionality which lets us view the different parts of the model at different levels of detail.

There are 2 methods to drill-down (or up):

Expand/Collapse – elements in the model which contain other elements, display a small  (expand button) or  (collapse button). Clicking on the expand/collapse button will cause the model editor to display the contents of the element (expand), or hide them (collapse).



Double-clicking an element in the model expands it to display its contents, and in addition causes the editor to center the view on the model and zoom in or out so the model fits in the view.

Zoom-in/out

There are various methods for zooming in and out:

Double-clicking an element in the model editor will cause the editor to zoom in/out and center on the element (in addition to expand if applicable).

Double-clicking an element in the outline is similar to double clicking in the editor. This method is useful when we want to move directly to part of the model which is not in the current scope of the display.

The toolbar provides buttons for zoom-in () and out ()

Zoom works slightly differ for data-elements. Double-clicking will only function when performed on the top data element, and not on any of its descendant data elements.

d) The Repository Explorer

The Repository Explorer, which appears to the left of the model editor, provides a complete list of all the models making up an application as shown in figure 12.4.

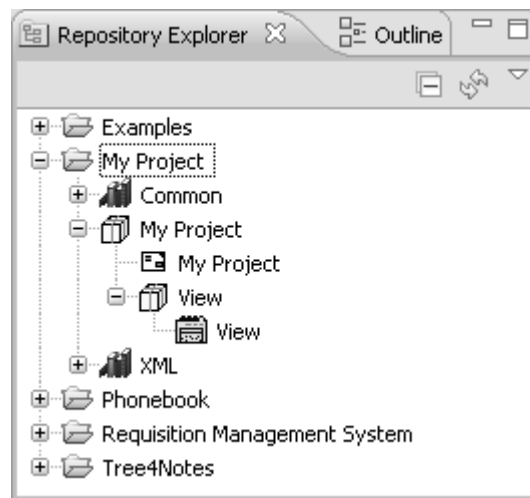


Fig. 12.4: Repository explorer

Finding your way in the Repository: The repository is organized hierarchically into Packages (and sub-packages), which group the models into functional categories. Packages are created automatically when certain templates (System, View, Button) are used, but you can always create additional packages and move models from one package to another to organize them as you see fit. When modeling, new elements are automatically created in the same package as the parent model to which they are added.

In order to locate, in the repository, the model you are currently editing, you can use the following shortcut:

- Right-click the element in the editor.
- Select the Show in Repository Explorer option from the context menu.


e) Embedded Application and Database Servers

Tersus includes a bundled, lightweight Application & Database Server, which can be used to view and test the modeled application at each stage of the modeling process.

The embedded servers are controlled through the studio's toolbar:




Launch the application, in the application server and open the browser.

 Stop the application, from running in the application server.

 Show the application log file in a text editor window.

Try to view your application, as follows:

Click the  button to start the application in the server, and open it in a browser.

You can switch back to the studio, change your modeling, and after saving your changes, if you switch back to the browser it will refresh automatically to apply your latest changes to the application model and database structures.

12.4.2 Tersus model libraries

Model libraries consists of Common Data Types, are predefined data items and template data structure templates which serve as building blocks for creating data elements.

The library contains the following categories of data types:

Data Types, Basic Data Types, these category contains both atomic data types and composite data types.

Atomic data type: The atomic data types are all reusable data types:

Text – A textual data item (string)

Number – A numeric data item (for any type of a number)

Date – A date data item

Date and Time – A 'date and time' data item

Boolean – A Boolean data item (yes/no)

Binary – A binary data item (byte buffer), used mainly to handle the contents of files

Anything – A generic data type used to indicate a data element with varying data types. For example, the Chooser template includes <Value> and <Options> data elements of type anything. This allows the elements to be compatible without having to explicitly specify the type of both.

Nothing – A generic (type less) data type used to indicate a slot that does not pass data. For example, the Find template includes a <None> exit of

type Nothing, because by definition it outputs no data (and only serves as the source of flows that activate subsequent actions).

Composite data type

The composite data types are either reusable predefined data structures or container templates for defining new composite data types:

Data Structure – An empty container for defining a composite data type. The data structure is composed of other atomic and/or composite data elements.

Database Record – An empty container for defining a composite data type that are mapped to a database table. The data structure is composed of other atomic and/or composite data elements.

Map – A collection of key/value pairs. Can be used to store the content of data structures with variable structure.

File – A predefined data structure for handling files over HTTP connections. The File data structure is used by various file handling templates, such as File Input Field & Show File.

Error – A predefined data structure for handling errors and exceptions.

Local and sharing properties that are available in the Tersus model libraries helps in builds blocks for assembling applications.


12.4.3 Tersus Server

In order to deploy Tersus applications you must first set up a deployment environment which includes the following components:

- Java Runtime Environment (JRE) 6
- Application server with Java Servlet support - Apache Tomcat 6.0.x
- Database server - HSQLDB, PostgreSQL, MySQL (Commercial DBMSs such as Microsoft SQL Server, DB/2, Oracle are supported by our commercial enterprise version)
- Tersus Server

Once a deployment environment has been set up, you may deploy your tersus applications by copying your project from the studio environment to the server environment and creating an application specific context file.

Self Assessment Questions

6. Tersus platform developed for _____ community.
7. RDF stands for _____
8. The Tersus Studio uses the Eclipse platform. State {True/False}.
9.  used for _____
10. Name the two methods for drilldown or drill up.

12.5 Modeling the Basic Display

We will discuss the basic modeling display and the logic behind the screen with a simple application design.

To start up with we need to create new application project, for this select File -> New-> Tersus project. Tersus project window appears type the project your name against the project name label. Select the legacy navigation under the template and leave the theme as default. After this press finish button to complete the task.

The next step is to start the modeling in order to create a form to accept requisition. To start modeling we need to open the application root model in the model editor. Since the project is just created, Tutorial should already be open in the model editor. If not, do the following:

Locate the Tutorial project root (folder) in the Repository Explorer view, and double-click it.

This should open a new editor window, which should look as show in figure 12.5 representing the application. Since we have not modeled anything yet, the rectangle is empty, except its name.

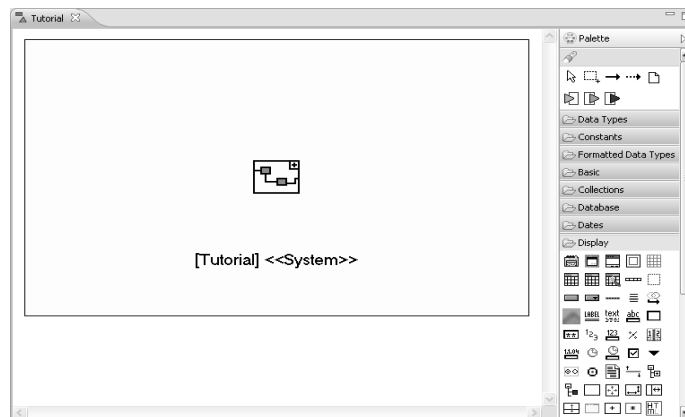



Fig. 12.5: Editor Window

First we need to define view model for an application consists of various displaying elements. To create a view, go to palette and check for the Display category is open and click on the view template  to select it. Now you can see the default view is open with the default name type open Requisitions and press enter to finish. After this window will appear as shown in figure 12.6.

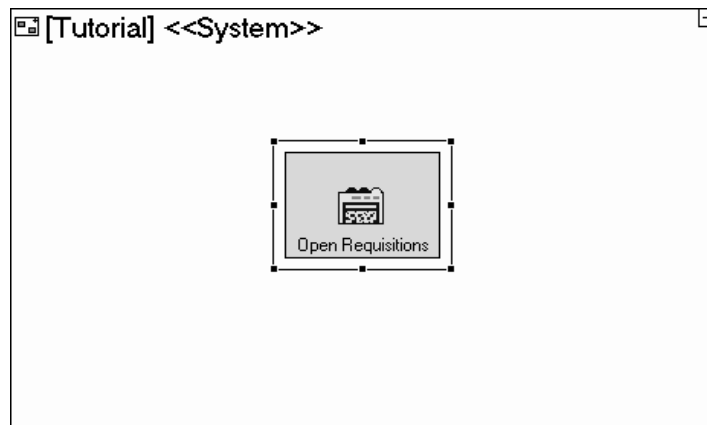


Fig. 12.6: Model window


The small rectangle inside is the sub model of the root model. Any model in the hierarchy may contain any number of sub models.

12.5.1 Modeling the logic behind the screen

Now you must have got an idea to model a form, the data what the user is going to enter needs to be stored and the code or the logic is required to achieve this task. As we discussed earlier Tersus modeling has the technique in which the logic is defined inside the display element that invokes it. Now we are going to design this requisition in our button click, this needs to be designed inside the submit button. Zoom into the Submit button by double-clicking on it in the model editor (or in the Outline view).

Defining data structure

We will now define a data structure to store the details of a requisition:

Select the Data Types/Database Record template  and drop it into the button. Name it "Requisition". Now the submit model will appear as shown in figure 12.7.

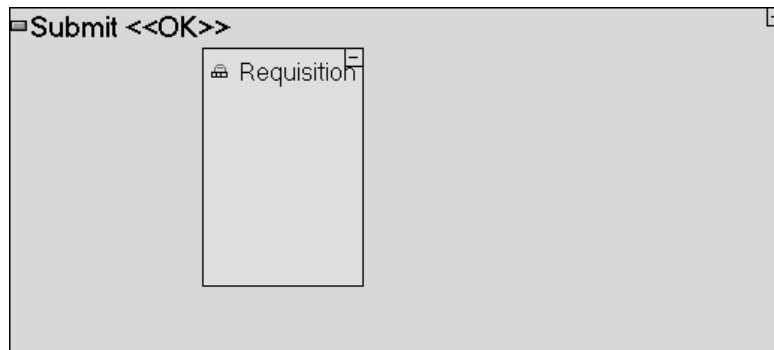




Fig. 12.7: Submit model


Next we should define the fields comprising the data structure. To insert data elements into the data structure, we simply drag the appropriate data types (Number, Text, etc.).


- Select the Data Types/Number template () and drop it into the Requisition data structure. Name it Id.
- Select the Data Types/Text template () and drop it into Requisition as well. Name it Description.

Now 2 fields are inserted in the Requisition data structure: Id, needs to be an automatically generated unique identifier for each requisition, and Description, which is a free text description of the requisition as entered by the user.

Process for unique identifier

Here Id must be a unique identifier that means it needs to be automatically generated. It must be unique, since we don't require any two requisitions to hold the same identifier. This can be achieved using the Sequence Number template:

Select the Database/Sequence Number template () and drop it into the Submit model. Name it Requisition Id.

The process nothing but the "Sequence Number" generates an output of unique nature. The output needs to "exit" the process in order to be used by another process or populate a field of a data structure. An Exit slot is modeled as a gray triangle () that is available on process frame model. The unique identifier is generated by the sequence number template that are created through the predefined <Next> exit.

Create a Flow

Of course, we still need to pass this generated identifier from the <Next> exit slot (➡) of Requisition Id to the Id field of the Requisition data structure. To do this we shall use a Flow.

Click the Flow tool (→) at the top row of the palette, click on the <Next> exit slot (➡) of Requisition Id to specify the Source, and then click on the Id field in Requisition to specify the Target. An arrow should appear by binding or creating a link between the field with the slot. A Flow is modeled by an arrow between Source and Target model elements (). This can represent the relationship between the slot and the field in the following two ways

- Order of Execution: On what condition the process needs to be executed under which sequence.
- Data Flow: when and how the data items can be passed between.

Figure 12.8 depicts the model appears with sequence number and flow.

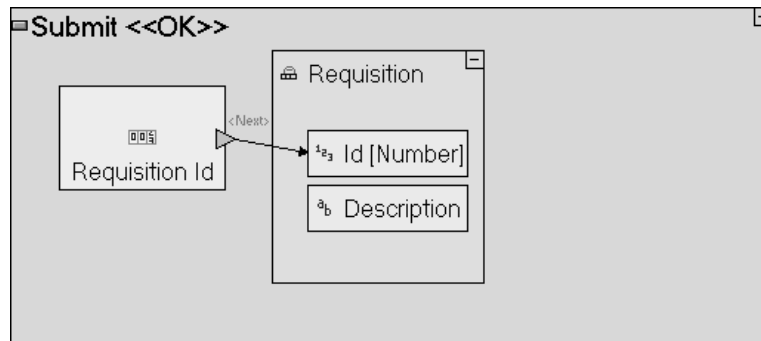


Fig. 12.8: Model with sequence number and flow

Display data element

So now we automatically generated the identifiers in the Requisition data structure; we need to store the Description in the same data structure. To achieve this we shall define a new type of element in the model called a Display Data Element.

A “Display Data Element” is a substitution representation of a Display element in the form of data structure. The goal of this access provision to content is to read or write data. Its main idea is to provide access to the contents of the display so that it can be read from or written to for display purpose.

We would like to access the contents of the Description text area, and so we must add to the Submit button a Display Data Element that references the Enter New Requisition popup. Since Enter New Requisition is the “*father*” of the Submit button, we use the Add Ancestor Reference operation:

- Right-click on the Submit button, select Add Ancestor Reference from the menu, and select Enter New Requisition.

Next you need to create a flow from the Description text area within Enter New Requisition to the Description field in the Requisition Data structure. The actual text entered by the user is available through the <Value> data element of the Description text area within Enter New Requisition data element:

- Use the Flow tool (→) to link Enter New Requisition/Description/ <Value> to Requisition/Description. Figure 12.9 shows the model with requisition display with submit button.

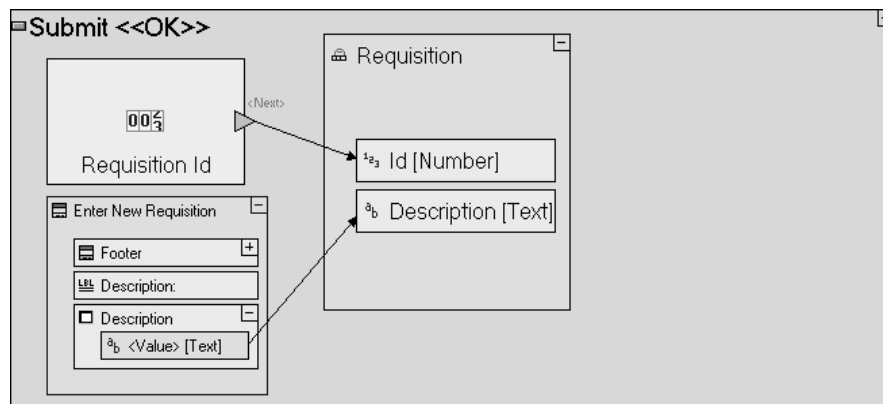


Fig. 12.9: Display data with Requisition

Thus the Requisition Id sub-process generates an identifier that is passed to the Id field of Requisition, and the content of the Description text area (entered by the user) is passed to the Description field of Requisition.

Self Assessment Questions

11. State the two controls that the flow can have over the data
12. `a_b` accepts _____ data type.

12.6 Summary

- Tersus is a dual licensed open source and proprietary visual programming platform for software development.
- The Tersus Modeling language supports in designing user interface, both server and client side processing
- Tersus IDE helps the programmers to design the application in a graphical environment.
- Tersus studio, model libraries and Tersus server are the three main components of Tersus platform.
- Tersus is a language independent platform it has built in templates and prototypes supports the programmer to design the application easily.

12.7 Terminal Questions

1. Discuss on the Tersus visual platform
2. Discuss the architecture of Tersus platform
3. Explain the features of Tersus platform
4. Describe the features of Tersus studio
5. Write the procedure to develop a simple application in the Tersus platform.

12.8 Answers

Self Assessment Questions

1. Tersus studio
2. Flow.
3. False
4. Web Service Definition Language.
5. Log file.
6. OSS
7. Resource Description Framework
8. True
9. Adds note for an existing element
10. Expand and collapse.
11. Order of Execution & Data flow
12. Types or Text template

Terminal Questions

1. Tersus is a dual licensed open source and proprietary visual programming platform for software development. It supports in application development especially for web based application. For more details refer section 12.2.
2. It consist of modules, GUI components, activity units, data types etc. For more details refer section 12.2.
3. Language independence, importing services, common libraries built in templates and prototypes are the various features of Tersus platform. For more details refer section 12.3.
4. Tersus Studio, supports with multiple tools to support modeling, such as the Model Editor, Outline, Repository Explorer etc for more details refer section 12.4.
5. Steps in developing applications are load a design window, include submit button, process for unique identifier, flow for binding control depicts the role of data flow and the display of data element. For more details refer section. 12.5.

E-Reference:

- <http://www.tersus.com/?Id=26>
- <http://infocenter.tersus.com/index.jsp?topic=/tersus.help/html/Tutorial/Appendix A/Appendix A.html>
- <http://en.wikipedia.org/wiki/Tersus>