

SWIN
BUR
NE

SWINBURNE
UNIVERSITY OF
TECHNOLOGY

Automatic Diagram Layout Support for the Marama Meta-toolset

Pei Shan Yap and John Hosking,
University of Auckland,

New Zealand

John Grundy, Swinburne University
of Technology, Australia



Outline



- Need for automatic layout
- Marama meta-tools
- Adding specification & generation of auto-layout to Marama meta-tools
- Example
- Future research

Need for automatic layout



■ Example 1: MaramaEML enterprise modelling tool

The screenshot displays the Eclipse SDK interface for the MaramaEML enterprise modelling tool. The main workspace is divided into several panes:

- (a)** A hierarchical service diagram showing the structure of the University Enrollment Service, including Student Service, Enrollment Office, and Department, with various tasks and connectors.
- (b)** A BPMN-style process flow diagram showing the execution flow between Student, Enrollment Office, and Department, with tasks like 'Apply Enr...', 'Receive...', 'Check Ac...', 'Approve...', and 'Report S...'. It includes annotations like '[1] I want to Enroll this Course' and '[5] Exam Time Impact?'.
- (c)** A toolbar with tools for Select, Marquee, Sketching tool, Fisheye Zoom, and Shapes.
- (d)** A list of EML elements such as EMLException, EMLProcessEnd, EMLProcessStart, EMLService, EMLNameLabel, Handle, EMLOperation, and Focus.
- (e)** A list of connectors including EMLException..., EMLIterativeFlow, EMLProcessFlow, EMLTrigger, TreeBranch, EMLNameLink, and Link.
- (f)** A Properties pane showing the configuration for a selected element, with fields like fillColor, id, inputValue, lineColor, lineVisible, Location, name, and outputVariable.
- (g)** A context menu with options like Collapse this service node, Expand this service node, Generate BPEL4WS from EML, Show EML Exception Flow, Show EML Process Flow, Show EML Trigger, Hide EML Exception Flow, Hide EML Trigger, Run As, Debug As, Team, Compare With, and Replace With.
- (h)** A Processes pane showing a list of process IDs and their corresponding process sequences.

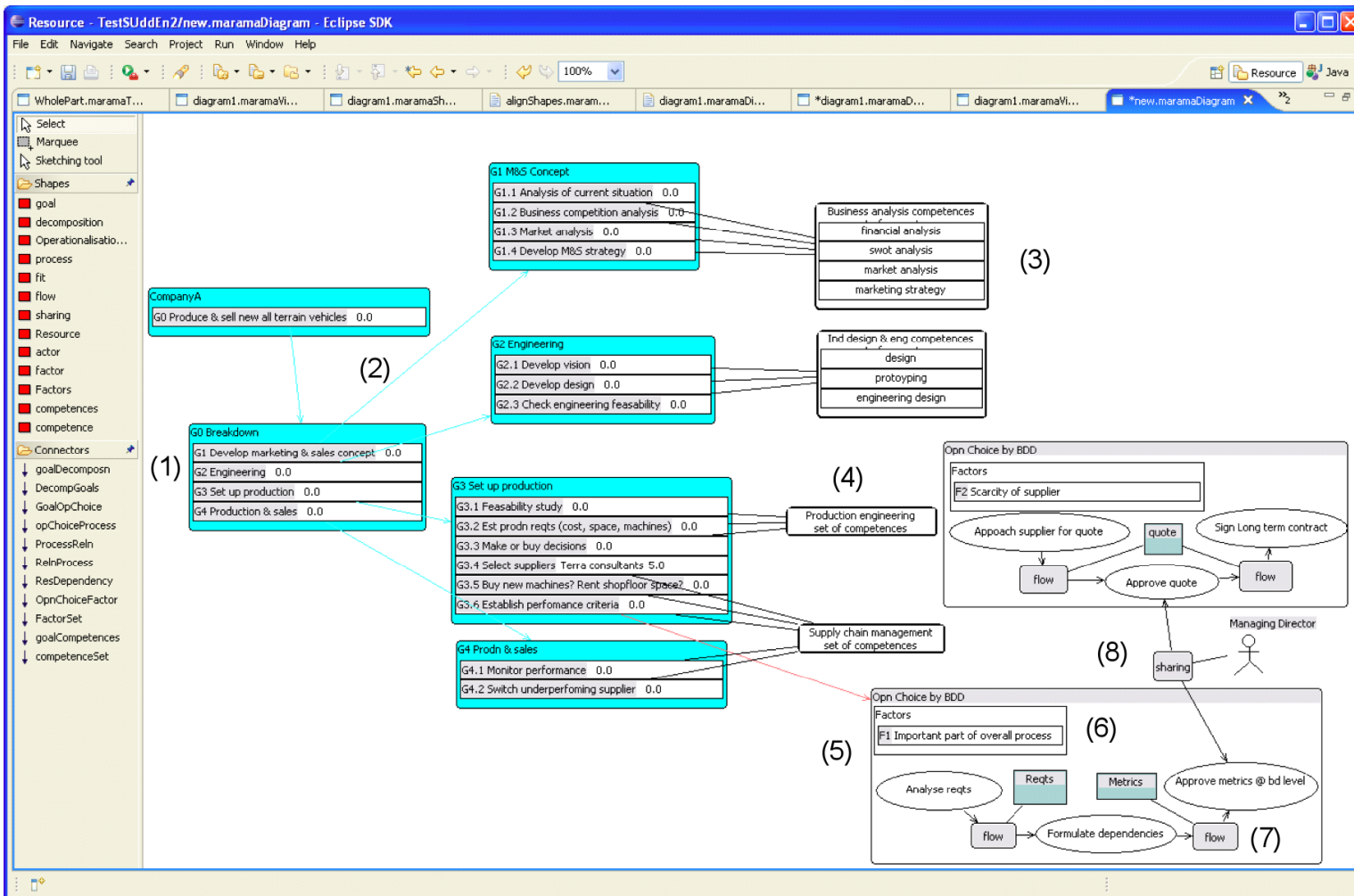
Property	Value
fillColor	RGB (204, 204, 204)
id	id
inputVariable	Student ID
lineColor	RGB (255, 255, 255)
lineVisible	true
Location	339, 346
name	name
outputVariable	Reference Number

process id	ProcessSequence
0	Show All Processes
1	Enroll a Course
2	Enquire Course Information
3	Change Enrollment
4	Drop the Course

Need for automatic layout



■ Example 2: MaramaSUDDEN supply chain modeller



Need for automatic layout



■ Example 3: Visual Wiki

The screenshot shows a web browser window with the URL <http://thinkpedia.cs.auckland.ac.nz/>. The page features a search bar and navigation links. The main content is a radial network diagram centered on 'University of Auckland'. The diagram includes nodes for 'Teacher', 'Vice-Chancellor', 'Academia', 'Education', 'University of Auckland', 'Higher education', 'Performance Based Research Fund', 'University of Oxford', 'Education', 'Social Tag', 'THE-QS World University Rankings', 'Association of Commonwealth Universities', 'Auckland', 'University of Cape Town', 'Facility', 'University of New Zealand', 'University of Auckland', 'Auckland College of Education', 'City campus', 'City', 'Natural Feature', 'Organization', 'Auckland, California, United States', 'Auckland University College, West Yorkshire, United Kingdom', 'Auckland College of Education', 'School of Education', 'Faculty of Education', 'Information Management', 'Company', 'Technology', 'Person', 'Industry Term', 'University of Auckland', 'Stuart McCutcheon', 'Wine', 'Radio Station', 'Published Medium', 'Country', 'Province Or State', 'New Zealand', 'New Mexico, United States', 'Position', 'Auckland University of Technology', and 'New Zealand'. A sidebar on the right contains a Wikipedia article snippet for 'University of Auckland' and a list of languages.

THINKPEDIA
Visually Explore Wikipedia

WIKIPEDIA
The Free Encyclopedia

Article Discussion Read Edit

University of Auckland

From Wikipedia, the free encyclopedia

Not to be confused with Auckland

The University of Auckland (Māori: Te Whare Wānanga o Tāmaki Makaurau) is New Zealand's largest university and the top-ranked New Zealand university in the THES - QS World University Rankings. Established in 1883 as a constituent college of the University of New Zealand, the university is now made up of eight faculties over six campuses, and has more than 39,000 students at April 2006.^[3] Over 1300 doctoral candidates were enrolled at the University of Auckland in 2004. It offers a wide range of programmes including Arts, Business, Education, Music, Teacher Training and Special Education.

Main page
Contents
Featured content
Current events
Random article

Interaction
About Wikipedia
Community portal
Recent changes
Contact Wikipedia
Donate to Wikipedia
Help

Toolbox
Print/export

Languages
Deutsch
Español
فارسی
Français
한국어
עברית
Македонски

Marama meta-tools



- Set of (mostly) visual specification tools for Domain-Specific Visual Language (DSVLs)
- Meta-model designer, shape designer, view type (diagram) designer, various behaviour designers
- Java API with dynamic Java code plug-ins (scripts) – “event handlers” – for complex behaviour, integration

Marama examples - MaramaEML



■ Specifying MaramaEML tool:

The screenshot displays the MaramaEML tool interface, which is used for specifying MaramaEML tools. The main workspace shows a diagram with various shapes and connectors. The diagram includes shapes like ActivityShape, GroupShape, CommentShape, GateShape, StartShape, Activity, FlowConnector, Comment, MessageConnector, Activity_Activity, GroupContainsProcess, AlignContainedShapes, and GenerateBPEL. The diagram is organized into a hierarchy, with ActivityShape and GroupShape at the top level, and their respective sub-shapes and connectors below them.

The interface includes several panels:

- Package Explorer:** Shows the project structure, including packages like e5, EML1, diagram1.marama, diagram2.maramaViewType, EML1.maramaToolModel, MaramaMTE, nz.ac.auckland.cs.marama.tools, and uml1.
- Properties:** Displays the properties of the selected element. The left panel shows properties for the selected element, and the right panel shows properties for the selected element's parent.
- Outline:** Shows the outline of the diagram, including ViewMapping, ViewConnector, MappingLink, ViewAssociation, ViewMapping, MappingLink, ViewShape, ViewEntity, and ViewMapping.
- Toolbars:** Includes toolbars for Select, Marquee, Sketching tool, VisualUserHand..., VisualEventHan..., ViewAssociation, ViewConnector, ViewEntity, ViewShape, ViewMapping, Focus, Formula, Connectors, MappingLink, and FormulaLink.
- Bottom Panel:** Shows the MetaElement Attributes and Exported Icon Properties. The MetaElement Attributes table has columns for ID and name, with values ActivityID and activityName. The Exported Icon Properties table has columns for ID and name, with values ActivityID and activityName.

Specifying layout...



The screenshot displays a software development tool interface with several key components:

- Top Panel:** Contains menu items like "RemoveShapeAndConnector", "PositionNewTask", "NewConnectorConstraints", "ResizeConstraints", and "SnapToPlace".
- Left Panel:** A "Please specify the events this visual handler will respond to" dialog with checkboxes for "NewShapeEvent", "RemoveConnector", and "ChangeProperty". Below it, there are sections for "Please import any...", "Please input the...", and "Please briefly describe...".
- Navigator:** Shows a tree view with folders "e5" and "e6".
- Diagram Editor:** The main workspace shows a diagram with shapes like "ActivityShape" (green), "GroupShape" (green), "FlowConnector" (pink), and "Element_Element" (pink). A "shapeMoved" event is highlighted in a blue box. A red circle highlights a specific area of the diagram.
- Shapes and Connectors Lists:** Two lists on the right side show available shapes (ActivityShape, GateShape, StartShape, Commentshape, GroupShape) and connectors (FlowConnector, MessageConnector, GroupContainer).
- Workflow Diagram:** A separate diagram on the right shows a "ProcessEnrolment" flow with steps: "receiveApplication", "check complete", "check valid", and "issue OK". A "dept check" box is also visible.
- Bottom Panel:** A "snapToPlace" dialog with a "Select a formula:" dropdown and a text input field containing the formula: `encloses(GroupShape,ActivityShape,GroupContainerShape)`. This formula is circled in red.

Issues...



- Event handlers too complex/low-level for many Marama tool developers
- Kaitiaki (visual event handlers) also somewhat complicated to do e.g. force-directed, tree layouts
- View designer formulae require reusable event handler code and augmented shape descriptions (for layout control) – can't do without lots of manual augmentation; easy to make mistakes; very hard to change once done (high viscosity)

A new approach: MaramaALM



- Want to provide “minimal” augmentation of visual specifications to affect powerful layouts e.g. multi-level tree, force-directed, auto-layout
 - “I want this, this and this shape in a vertical tree layout for this view type...”
- Want to leverage internal Marama/Eclipse capabilities without any technical knowledge from tool developer
- Want specify-and-click to realise
- Want to be extensible approach
- Want minimal impact on Marama meta-tools vs additional meta-tool

Two-step specification



ExampleTool1.maramaToolModel | diagram1.maramaShapeType | diagram1.maramaViewType | diagram1.maramaDiagram

Property	Value
LayoutType	tree
Location	forced
Parent	tree
Size	38, 35

Properties (b)

Property	Value
LayoutType	forced
Location	forced
Parent	tree
Size	38, 34

Shapes (a)

- LabelShape
- TextFieldShape
- TextAreaShape
- ShapeViewer
- LayoutManager
- ShapeShape

Step 2



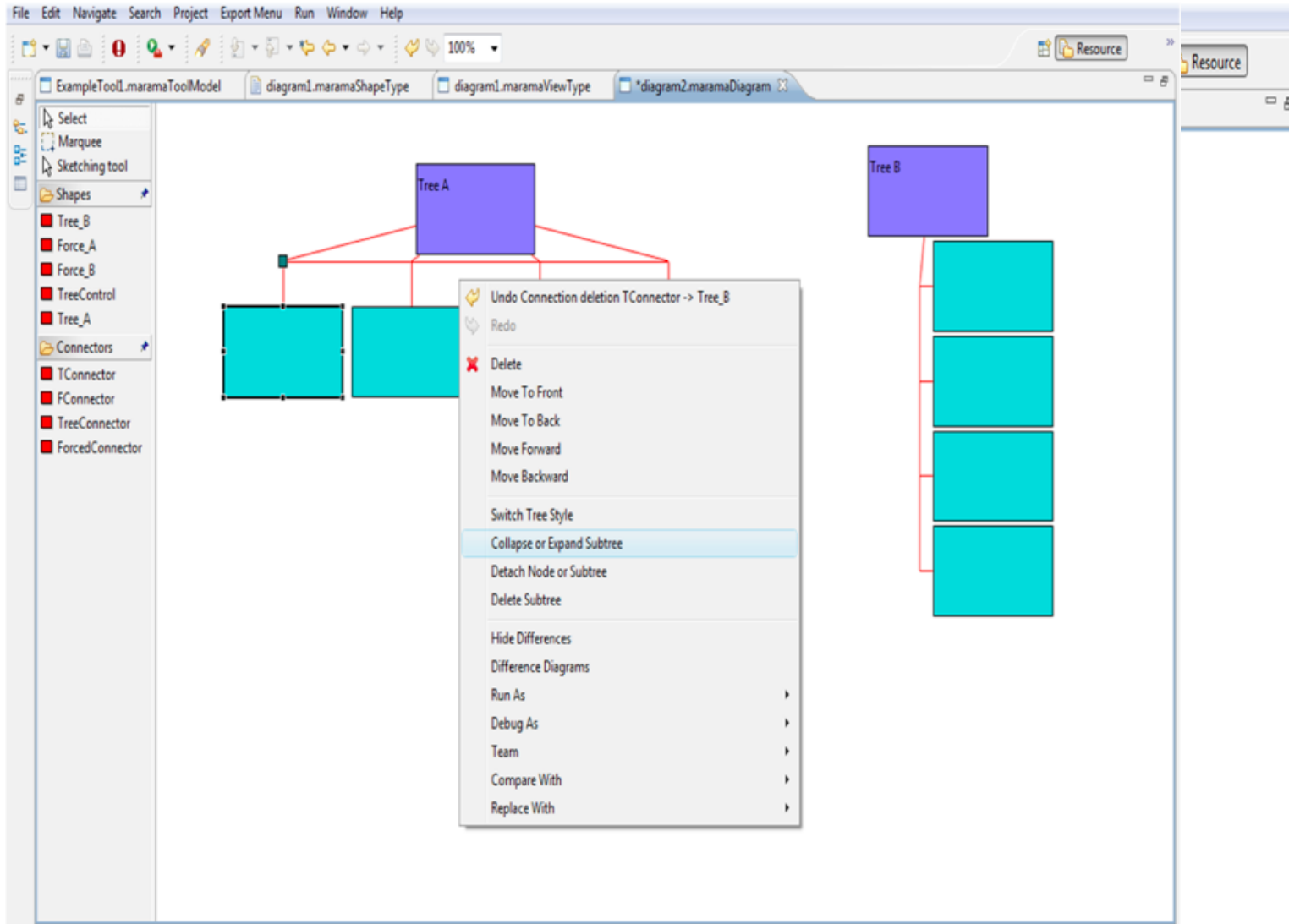
The screenshot shows a software interface with a menu open over a diagram. The menu is titled "Undo Move/resize ViewLayoutManager Rectangle(251, 379, 106, 67)" and contains the following options:

- Redo
- Delete
- Move To Front
- Move To Back
- Move Forward
- Move Backward
- Register the Viewtype
- Hide Formula Dependencies
- Generate Tree Layout Support
- Hide Differences
- Difference Diagrams
- Run As
- Debug As
- Team
- Compare With
- Replace With

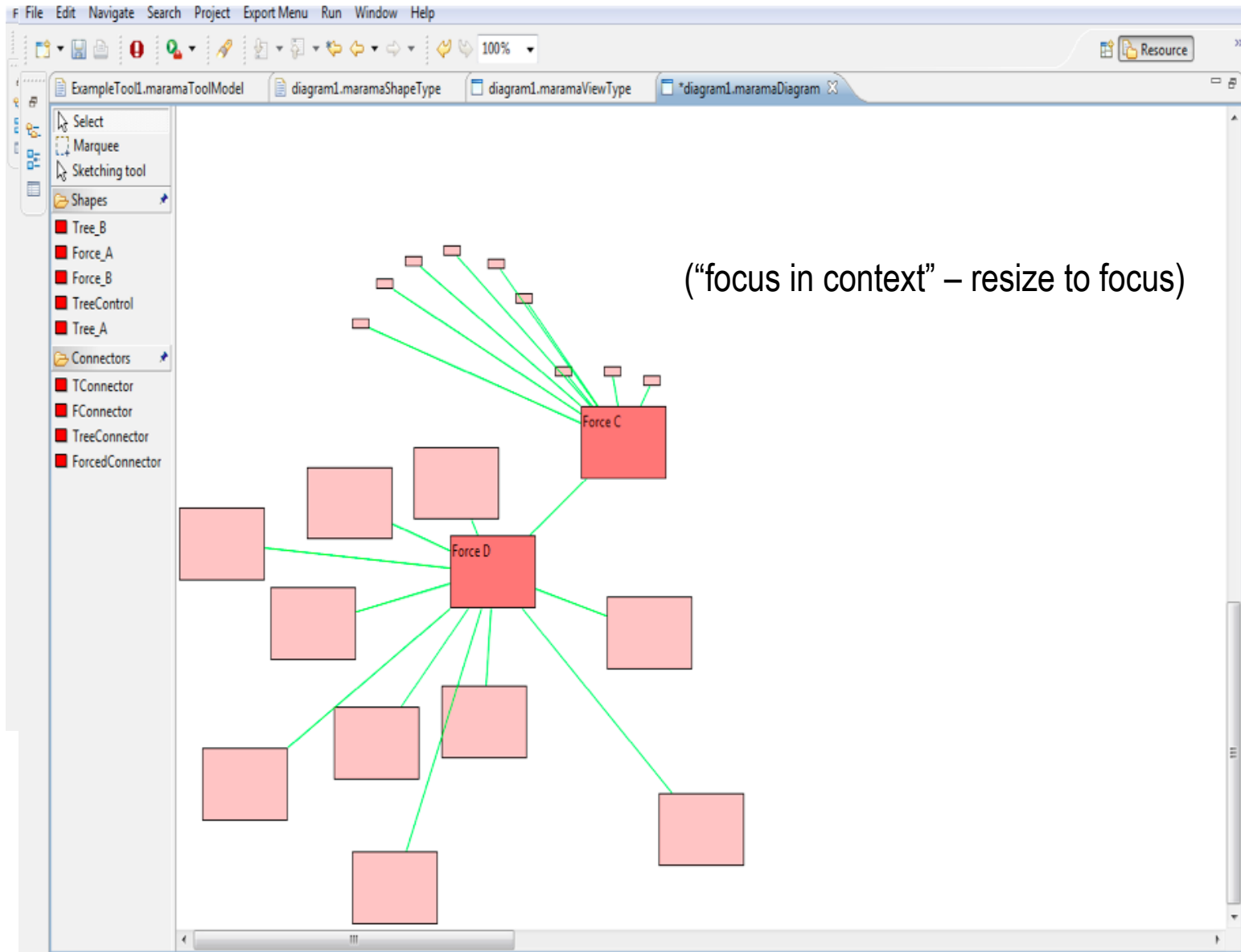
The diagram in the background consists of several nodes and connectors. The nodes are arranged in a grid-like structure. The top row contains nodes labeled "Tree_A", "Tree_B", "Force_A", "Force_B", "Tree_A_Tree_B", and "Force_A_Force...". The second row contains nodes labeled "Tree_A_Tree_A", "Tree_B_Tree_B", "Force_A_Force_A", "Force_B_Force_B", "TConnector_Tree...", and "FConnector_F...". The third row contains nodes labeled "Tree_A", "Tree_B", "Force_A", "Force_B", "TConnector", and "FConnector".

The interface also shows a toolbar with options like "Select", "Marquee", "Sketching tool", and "Shapes". The menu is currently open over the "Generate Force-Directed Layout Support" option.

Examples in use



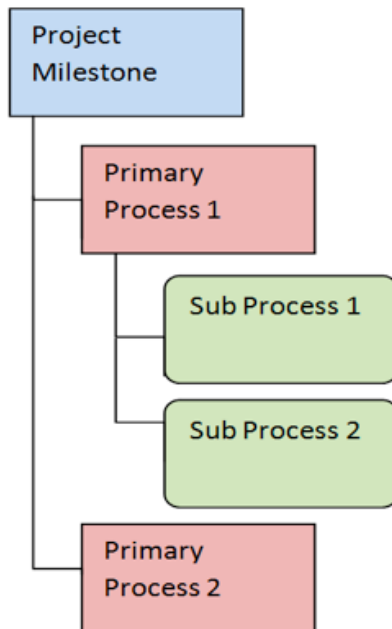
Example



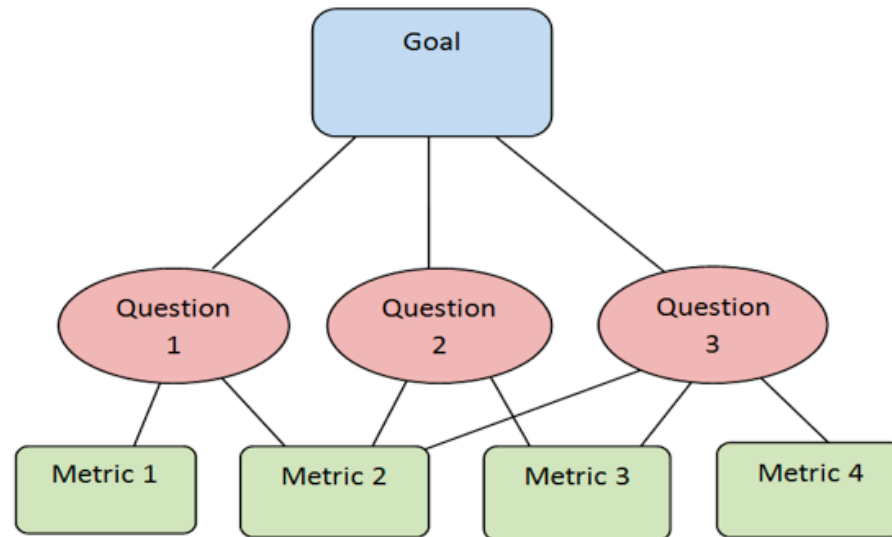
Case study: web metrics



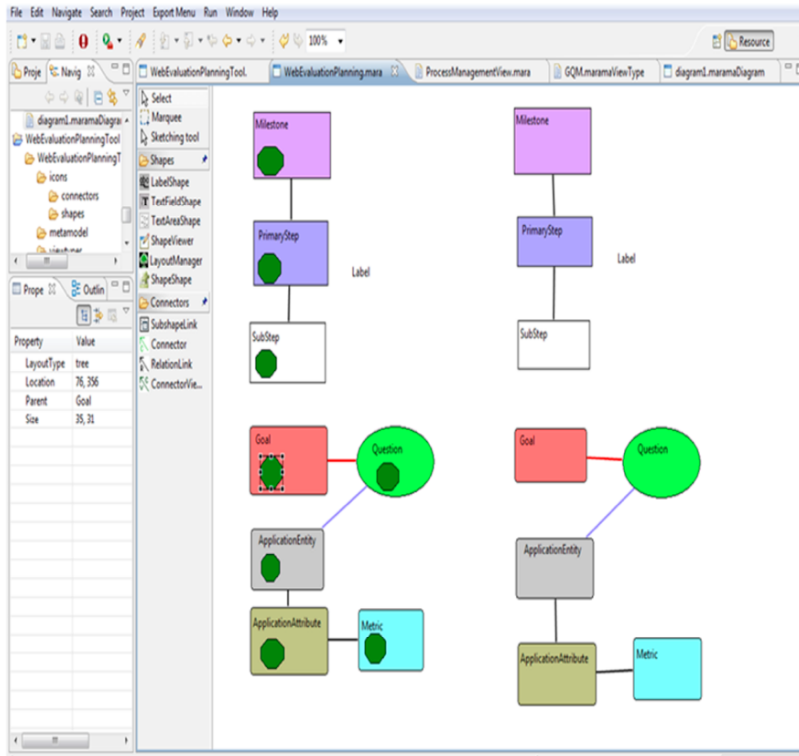
(a)



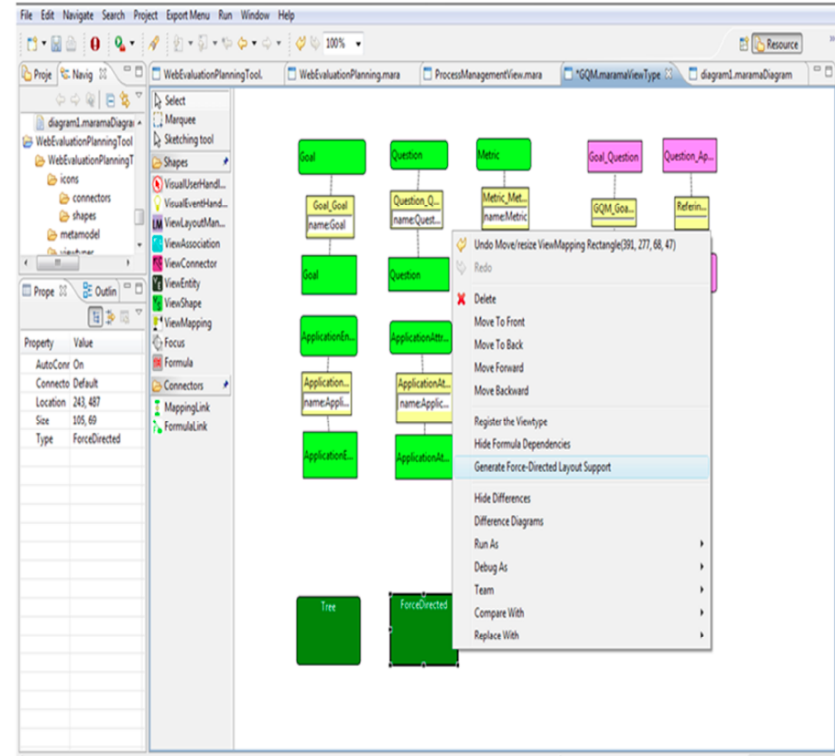
(b)



Specifying

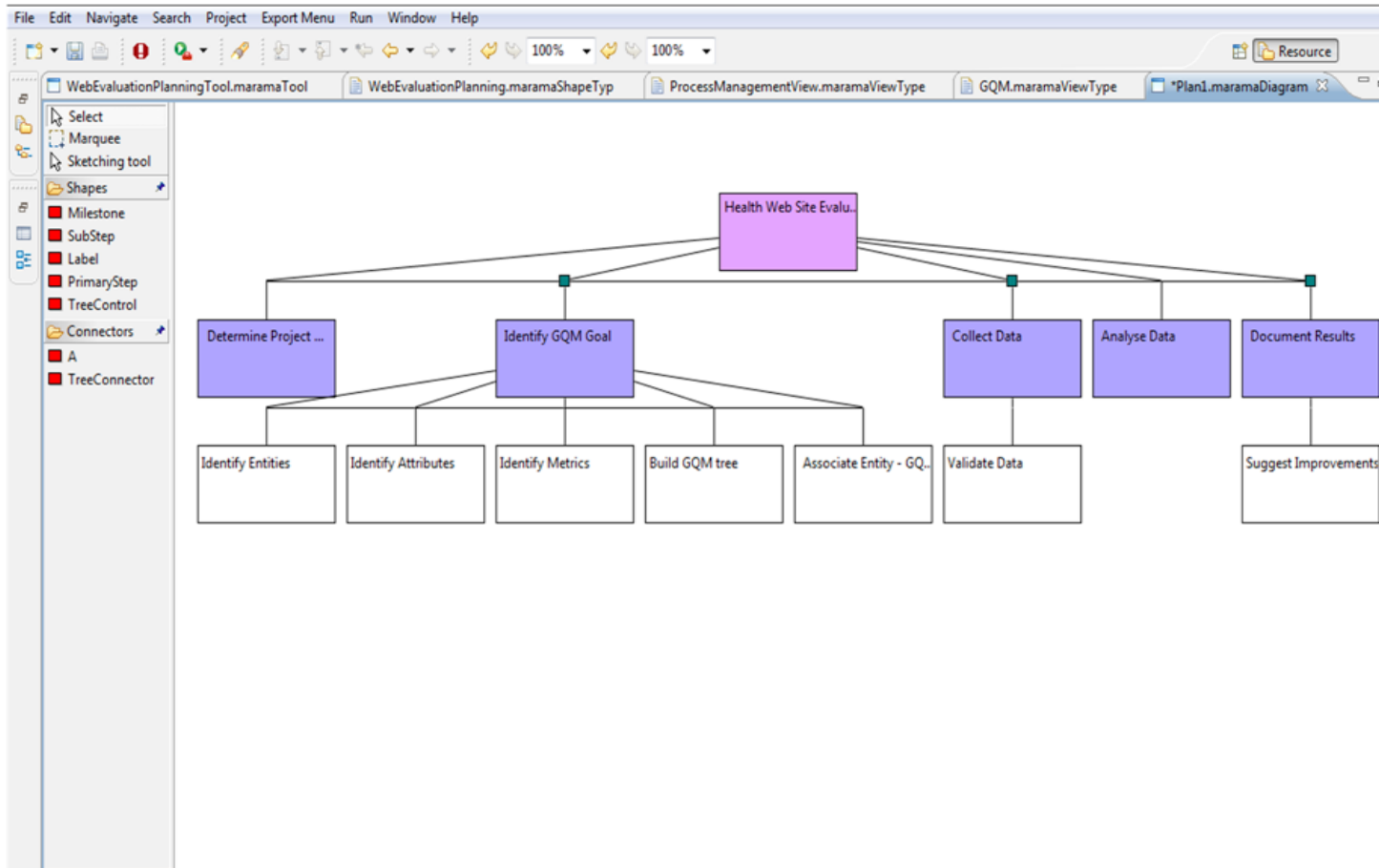


(a) Shape Designer

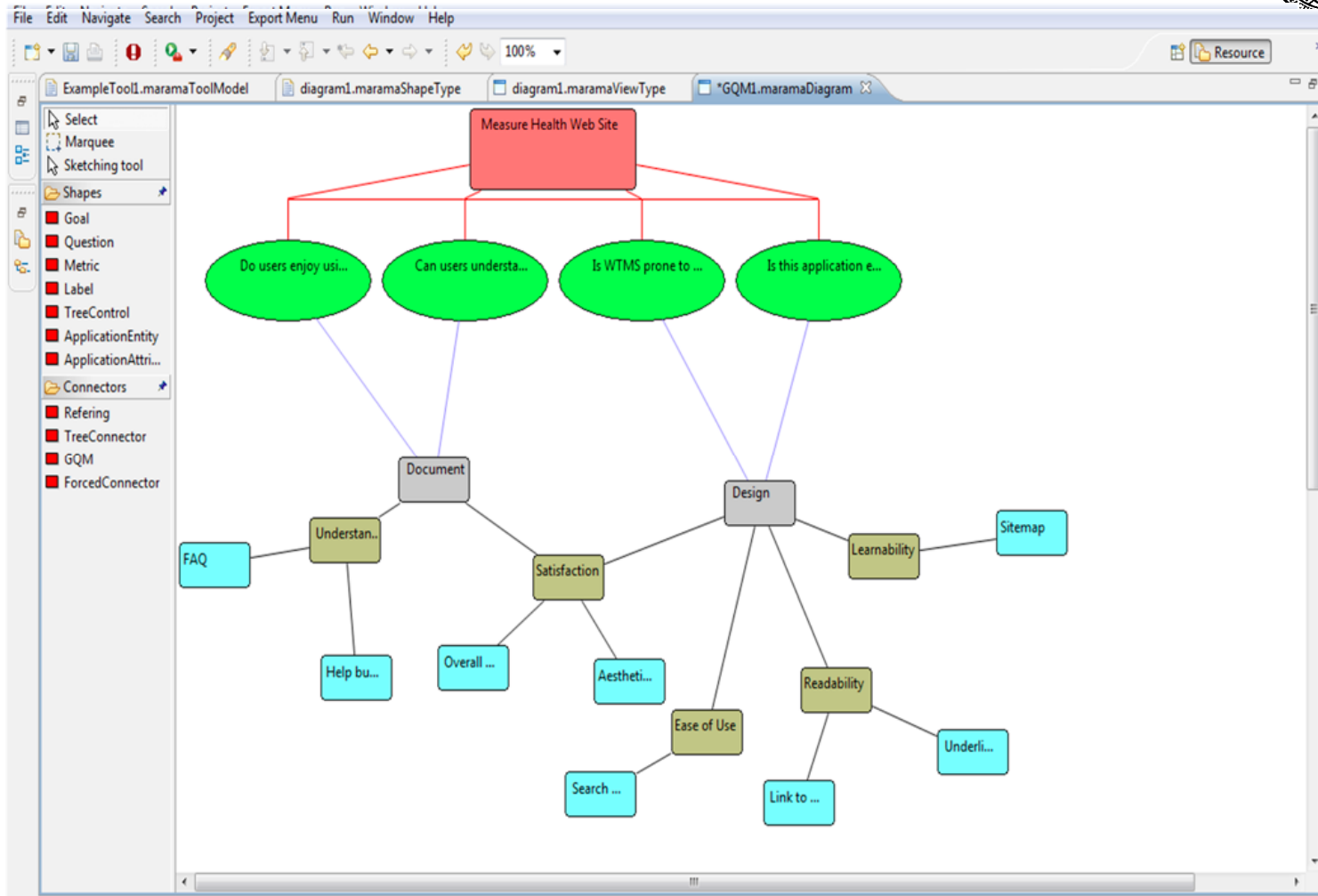


(b) View Definer

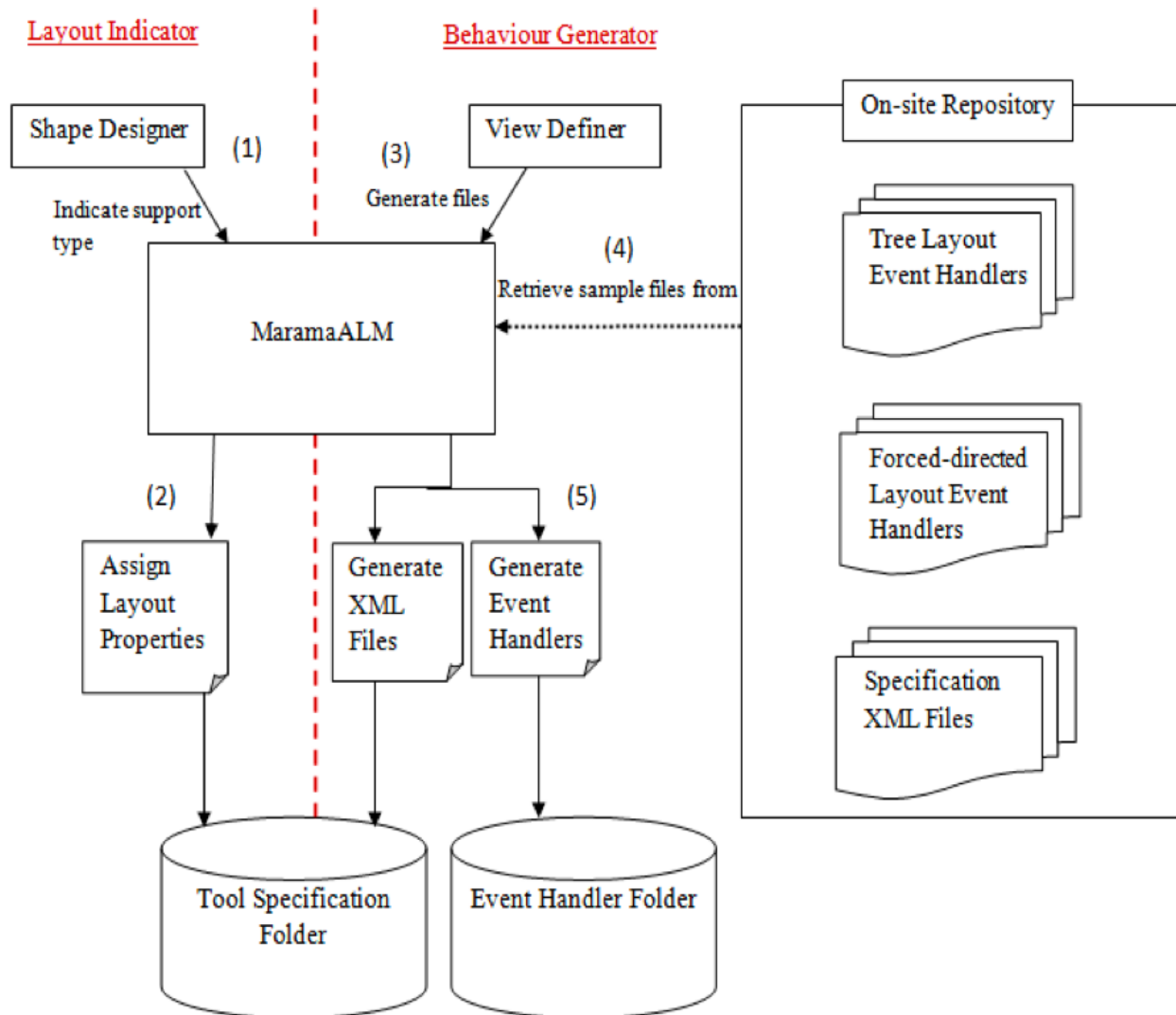
Using in generated tool



Using (2)



Architecture & Implementation



- Augmented the shape designer & view designer with additional meta-elements
- Set of layout handlers (Java)
- Augment shape, views from meta-elements – layout control properties, layout event handlers
- Generate augmented tool specifications

Evaluation



- Very fast to add to Marama tools; easy to change
- Auto-generates implementation in tool from visual specs
- Range of layout control for tool end user
- Can combine multiple layouts in same diagrams easily
- Limited layouts: trees and force-directed
- Very limited control of layouts (spacing, resizing etc)
- Can't specify new layouts directly
- Can't "see" the target layout in meta-tools – just annotations used to generate them

Summary / Future Work



- Extended Marama meta-tools with high-level layout control
- MaramaALM generates augmented tool specifications with additional properties & event handlers
- Usable by non-technical tool developers

- Add other layouts; reuse 3rd party layout libraries
- Allow meta-level specification of layout control
- Visualise the layout specifications better in meta-tools
- Add animation e.g. for force-directed layout

References



- Li, L, Grundy, J.C., Hosking, J.G. A visual language and environment for enterprise system modelling and automation, *Journal of Visual Languages and Computing*, vol. 25, no. 4, April 2014, Elsevier, pp. 253-277
- Grundy, J.C., Hosking, J.G., Li, N., Li, L., Ali, N.M., Huh, J. Generating Domain-Specific Visual Language Tools from Abstract Visual Specifications, *IEEE Transactions on Software Engineering*, vol. 39, no. 4, April 2013, pp. 487 - 515.
- Pei, Y.S., Hosking, J.G. and Grundy, J.C. Automatic Diagram Layout Support for the Marama Meta-toolset, In *Proceedings of the 2011 IEEE Symposium on Visual Languages and Human-Centric Computing*, Pittsburgh, USA, Sept 18-22 2011, IEEE Press.