magicdraw™

Architecture Made Simple

TOGAF PLUGIN

version 17.0.1

user guide

No Magic, Inc.
2011

CONTENTS

1 GETTING STARTED

TOGAF (Open Group Architecture Framework) is an open standard providing a comprehensive approach to the design, planning, implementation, and governance of enterprise information architectures.

In order to implement this standard, No Magic has developed and introduces for MagicDraw users a new free TOGAF plugin which provides:

- Fully featured enterprise architecture metamodel for content
- TOGAF diagram for content representation
- TOGAF project template
- Sample project

Chapter contains the following sections:

- Installing TOGAF Plugin
- Creating TOGAF Project
- Using TOGAF Sample

1.1 Installing TOGAF Plugin

There are several ways to install the TOGAF plugin. Choose one of the following:

- Download and install the plugin directly via the **Resource/Plugin Manager** dialog in the MagicDraw application.
- Download the *TOGAF_Plugin_<version number>.zip* file and then install the plugin via the **Resource/Plugin Manager** dialog.
- Install the plugin manually, in case the direct downloading and installing via the **Resource/Plugin Manager** dialog is not available.

To download and install the TOGAF plugin via the **Resource/Plugin Manager** dialog:

1. From the MagicDraw main menu, select **Help** > **Resource/Plugin Manager**. The **Resource/ Plugin Manager** dialog will open and prompt you to check for the latest product updates and resources. Click **Check for Updates** > **Check**.

   NOTE Specify HTTP Proxy Settings for the connection to start MagicDraw UML updates and resources.

2. Select the check box near the **TOGAF Plugin** and click **Download/Install**.
3. Restart MagicDraw.

To install the TOGAF plugin from the downloaded *.zip* file via the **Resource/Plugin Manager** dialog:

1. From the MagicDraw main menu, select **Help** > **Resource/Plugin Manager**.
2. Click the **Import** button to specify the *TOGAF_Plugin_<version number>.zip* file location. The plugin will be extracted and installed automatically.
3. Restart MagicDraw.

To install the TOGAF plugin manually:

NOTE Quit MagicDraw before installing the plugin.

1. Download the *TOGAF_Plugin_<version number>.zip* file.
2. Extract the downloaded file to the same directory wherein MagicDraw is installed.

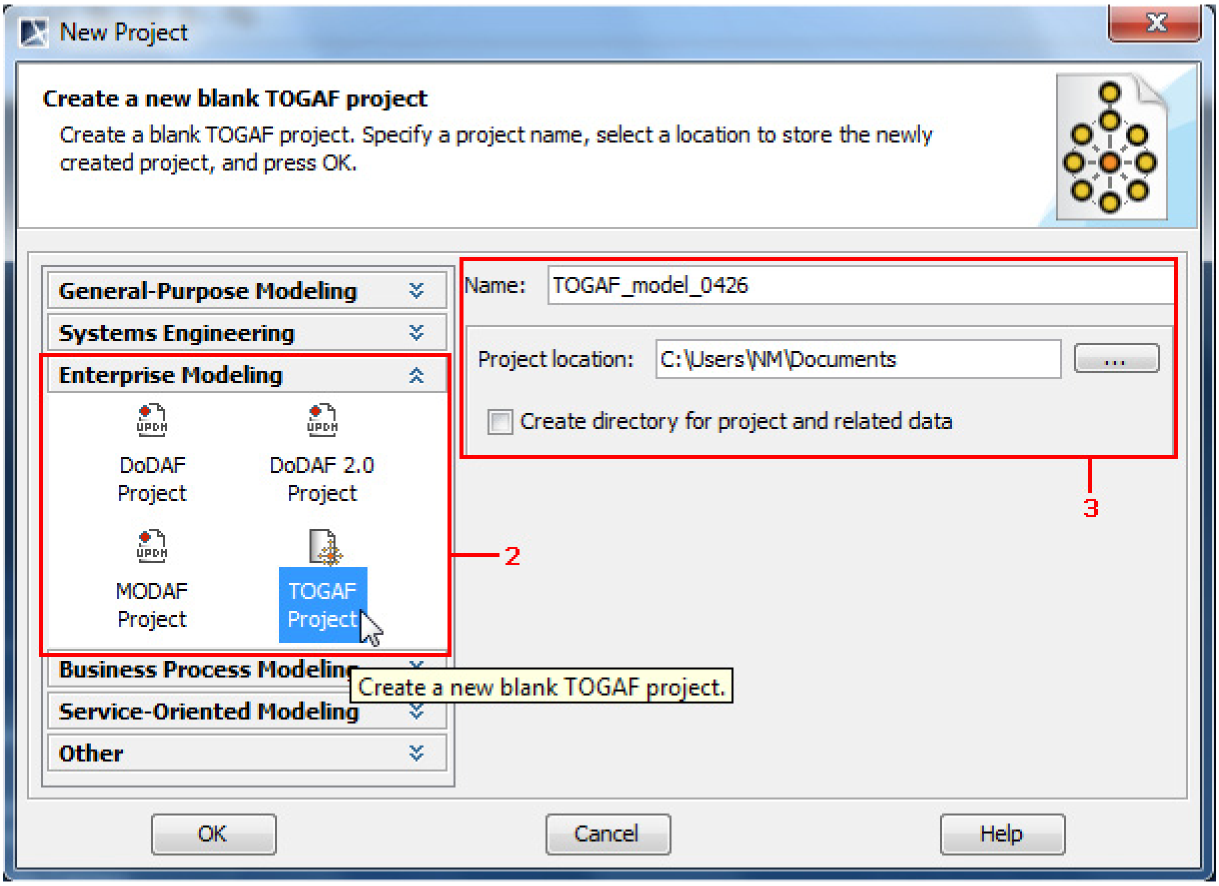
   IMPORTANT! If your operating system is other than Windows, use the command-line prompt to go to the MagicDraw UML folder and extract the *TOGAF_Plugin_<version number>.zip* file there.

3. Start MagicDraw. The TOGAF plugin is now applied to MagicDraw.

1.2 Creating TOGAF Project

To create a TOGAF project:

1. Do one of the following:
   - Under **Manage Project** on the left of the Welcome screen, click **Create New Project**.
   - From the MagicDraw main menu, select **File** > **New Project**.
2. In the **Enterprise Modelling** domain (you may need to expand it) on the left, click the TOGAF Project icon.
3. Specify project name and select location.
4. Click **OK**.



New Project

Create a new blank TOGAF project

Create a blank TOGAF project. Specify a project name, select a location to store the newly created project, and press OK.

Name: TOGAF_model_0426

Project location: C:\Users\NM\Documents

□ Create directory for project and related data

Figure 1 -- Creating new TOGAF project

1.3 Using TOGAF Sample

The TOGAF sample project gives an example of how to use the TOGAF Architecture Development Method (ADM) for the TOGAF-based enterprise modelling.

The TOGAF sample project is located in <MagicDraw installation directory>\samples\TOGAF.

To open the TOGAF sample project:

- From the MagicDraw main menu, select **File** > **Open Project**.
- Under **Manage Project** on the left of the Welcome screen, click **Open Project**.
- On the Welcome screen, click **Samples** and then under **TOGAF**, click **TOGAF sample**.

In the opened sample project, a content diagram with name “Index” is displayed. This diagram displays all TOGAF AMD phases and relations among them. Double-click each phase shape to open its sample diagram.

NOTE	Phases E, F, G, and H have no sample diagrams, because there are no particular guidelines for creating the model.

2 TOGAF DIAGRAMS

The following figure displays the process of creating a TOGAF Archirecture Development Method (ADM)-based model.

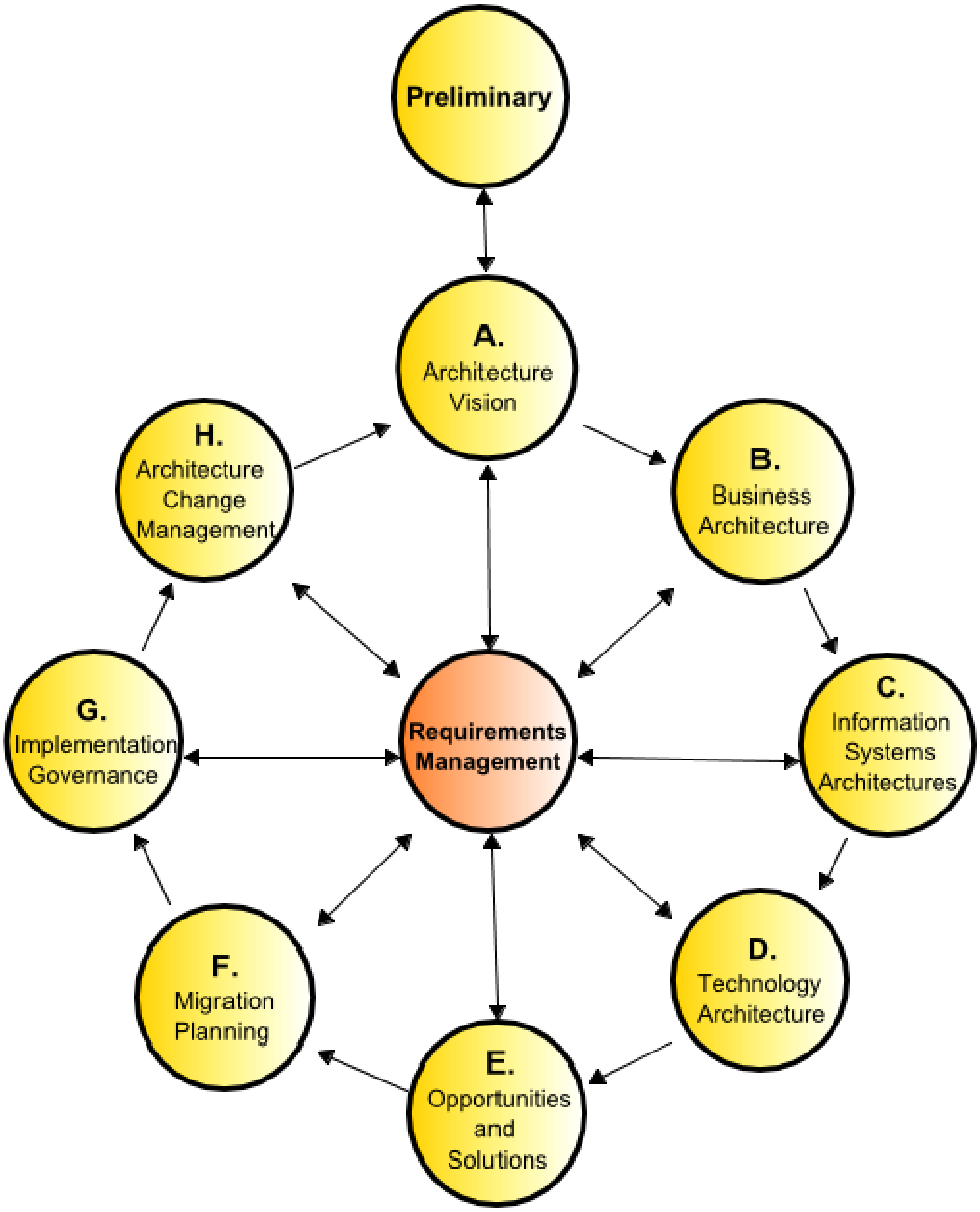

Preliminary
A. Architecture Vision
B. Business Architecture
C. Information Systems Architectures
D. Technology Architecture
E. Opportunities and Solutions
F. Migration Planning
G. Implementation Governance
H. Architecture Change Management
Requirements Management

Figure 2 -- TOGAF Architecture Development Method

Phase	Phase name	Description	Diagrams
	Preliminary	Preliminary defines the architecture principles that will form part of the constraints on any architecture work undertaken in the enterprise.	Preliminary diagram

Phase	Phase name	Description	Diagrams
A	Architecture Vision	Architecture Vision is the initial phase of the ADM. It includes information about defining the scope, identifying the stakeholders, creating the Architecture Vision, and obtaining approvals.	Architecture Vision diagram
B	Business Architecture	Business Architecture supports an agreed Architecture Vision.	Business Architecture diagram
C	Information Systems Architectures	Information Systems Architecture focuses on identifying and defining the applications and data considerations that support an enterprise's Business Architecture; for example, by defining views that relate to information, knowledge, application services, etc.	Applications Architecture diagram Data Architecture diagram
D	Technology Architecture	Technology Architecture will define baseline and target views of the technology portfolio, detailing the roadmap towards the Target Architecture, and to identify key work packages in the roadmap.	Technology Architecture diagram
E	Opportunities and Solutions	Opportunities and Solutions describes the process of identifying delivery vehicles (projects, programs, or portfolios) that effectively deliver the Target Architecture.	There are no particular guidelines for creating the model.
F	Migration Planning	Migration Planning focuses on creation of a viable Implementation and Migration Plan in co-operation with the portfolio and project managers.	There are no particular guidelines for creating the model.
G	Implementation Governance	Implementation Governance provides an architectural oversight of the implementation.	There are no particular guidelines for creating the model.
H	Architecture Change Management	Architecture Change Management goal is to ensure that the architecture achieves its original target business value. This includes managing changes to the architecture in a cohesive and architected way.	There are no particular guidelines for creating the model.
	Requirements Management	Requirements Management define a process whereby requirements for enterprise architecture are identified, stored, and fed into and out of the relevant ADM phases	Requirements Management diagram

Related web resources

Part II - ADM. In *TOGAF 9*. Retrieved April 18, 2011, from http://pubs.opengroup.org/architecture/togaf9-doc/arch/toc-pt2.html

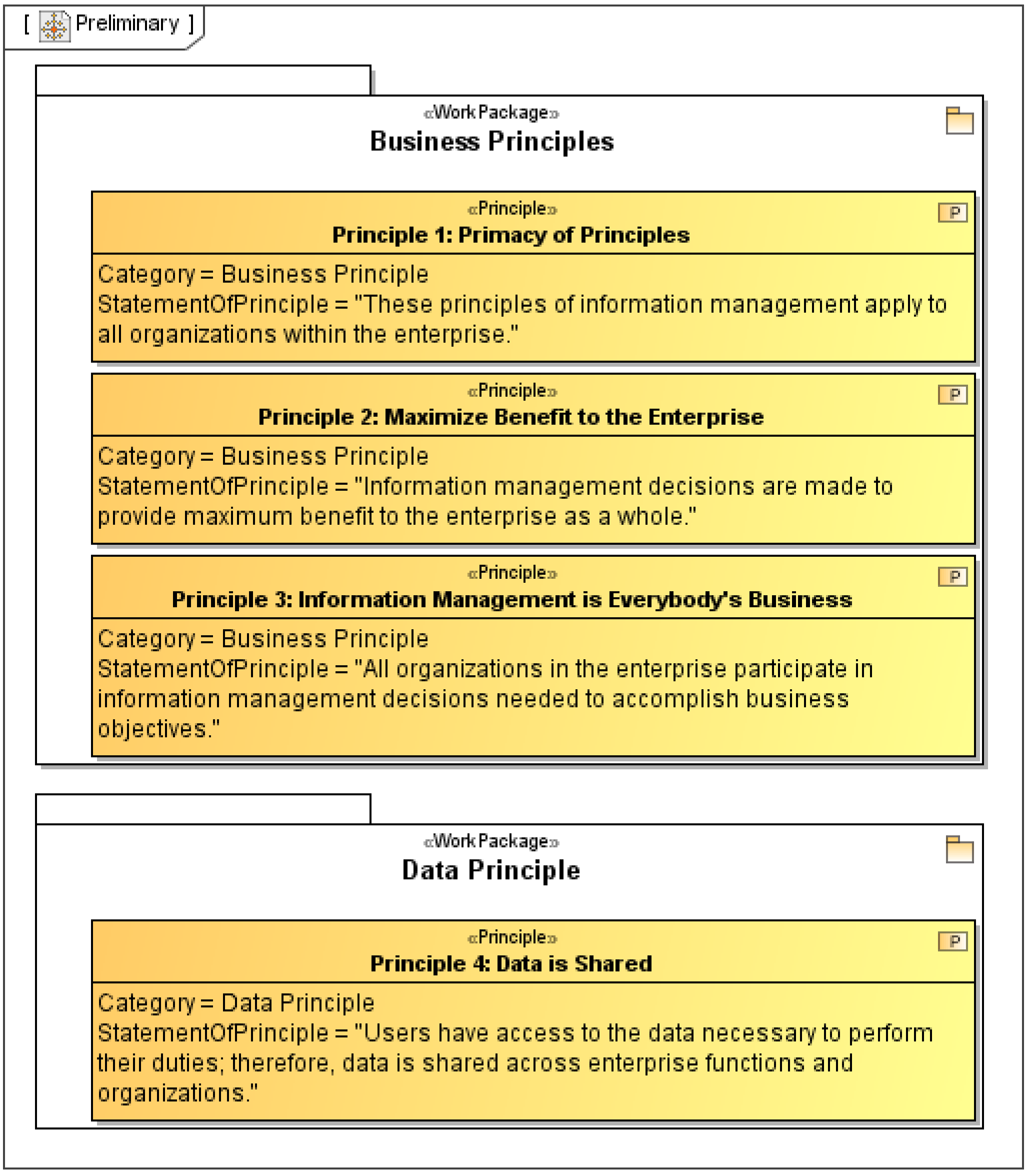
2.1 Preliminary diagram

Description

Preliminary phase defines “where, what, why, who, and how we do architecture” in the enterprise concerned. The main aspects are as follows:

- Defining the enterprise.
- Identifying key drivers and elements in the organizational context.
- Defining the requirements for architecture work.
- Defining the architecture principles that will inform any architecture work.

Sample



[Preliminary]

«Work Package»
Business Principles

«Principle»
Principle 1: Primacy of Principles
Category = Business Principle
StatementOfPrinciple = "These principles of information management apply to all organizations within the enterprise."

«Principle»
Principle 2: Maximize Benefit to the Enterprise
Category = Business Principle
StatementOfPrinciple = "Information management decisions are made to provide maximum benefit to the enterprise as a whole."

«Principle»
Principle 3: Information Management is Everybody's Business
Category = Business Principle
StatementOfPrinciple = "All organizations in the enterprise participate in information management decisions needed to accomplish business objectives."

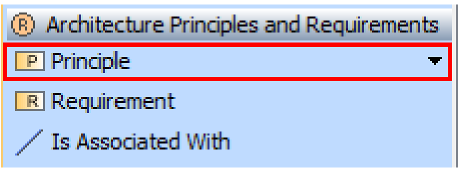
«Work Package»
Data Principle

«Principle»
Principle 4: Data is Shared
Category = Data Principle
StatementOfPrinciple = "Users have access to the data necessary to perform their duties; therefore, data is shared across enterprise functions and organizations."

Related procedures

Creating TOGAF diagram

Related elements



Principle

Available relations

Source element	Target element	Relation
Principle	Any element	⁄ Is Associated With

Related diagrams

Architecture Vision diagram

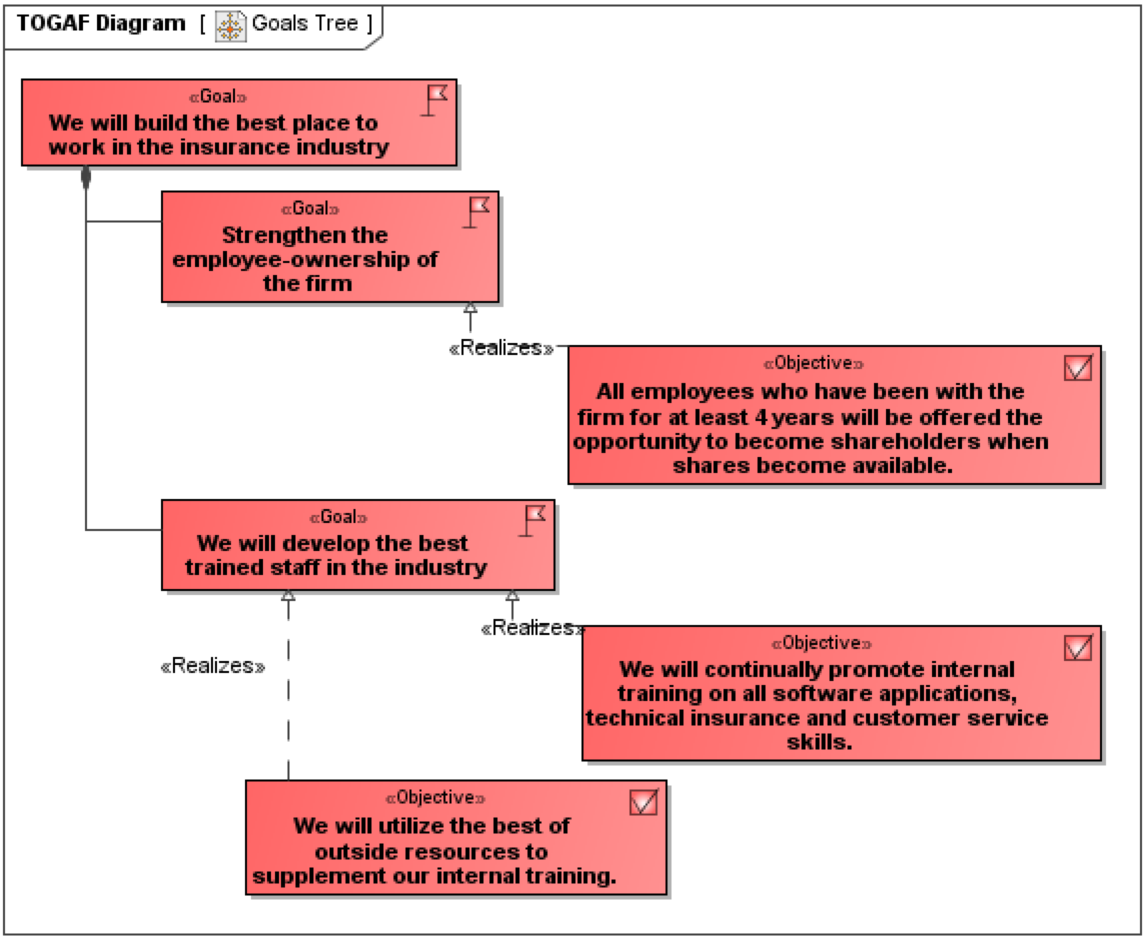
2.2 Architecture Vision diagram

Description

Architecture Vision defines what is in and what is outside the scope of the architecture effort and the constraints that must be dealt with. The constraints will normally be informed by the business principles and architecture principles, developed as part of the Preliminary phase.

Normally, the business principles, business goals, and strategic drivers of the organization are already defined elsewhere in the enterprise. If so, the activity in Architecture Vision diagram is involved with ensuring that existing definitions are current, and clarifying any areas of ambiguity. Otherwise, it involves defining these essential items for the first time.

Sample



We will build the best place to work in the insurance industry

Strengthen the employee-ownership of the firm

All employees who have been with the firm for at least 4 years will be offered the opportunity to become shareholders when shares become available.

We will develop the best trained staff in the industry

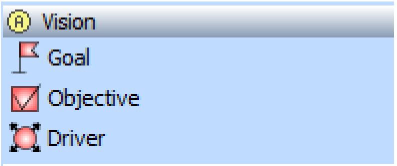
We will continually promote internal training on all software applications, technical insurance and customer service skills.

We will utilize the best of outside resources to supplement our internal training.

Related procedures

Creating TOGAF diagram

Related elements



Goal
Driver
Objective

Available relations

Source element	Target element	Relation
Goal	Goal	Decomposes
Objective	Objective	Decomposes
Objective	Goal	Realizes
Driver	Driver	Decomposes
Driver	Goal	Creates
Driver	Organization Unit	Motivates

Related diagrams

Preliminary diagram
Business Architecture diagram
Requirements Management diagram

2.3 Business Architecture diagram

Description

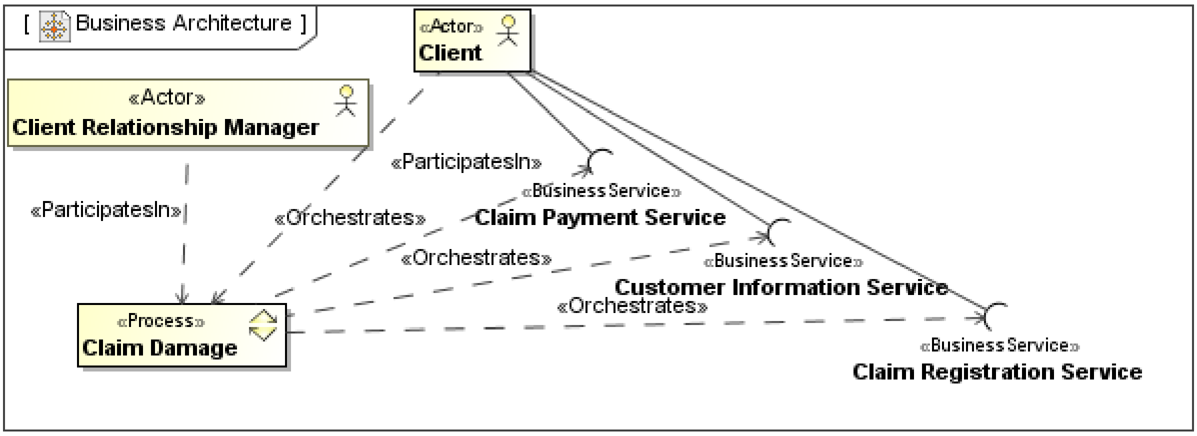
Business Architecture is often necessary as a means of demonstrating the business value of subsequent architecture work to key stakeholders, and the return on investment to those stakeholders from supporting and participating in the subsequent work. The role of the Business Architecture is to define how to achieve the goals and drivers, and the metrics for success. Research, verify, and gain buy-in to the key business objectives and processes that the architecture is to support.

Business models should be logical extensions of the business scenarios from the Architecture Vision, so that the architecture can be mapped from the high-level business requirements down to the more detailed ones.

A variety of modeling tools and techniques may be employed, if deemed appropriate. For example:

- Activity Models (also called Business Process Models) describe the functions associated with the enterprise's business activities, the data and/or information exchanged between activities (internal exchanges), and the data and/or information exchanged with other activities that are outside the scope of the model (external exchanges).
- Class Models are similar to logical data models. A class model describes static information and relationships between information. A class model also describes informational behaviors.

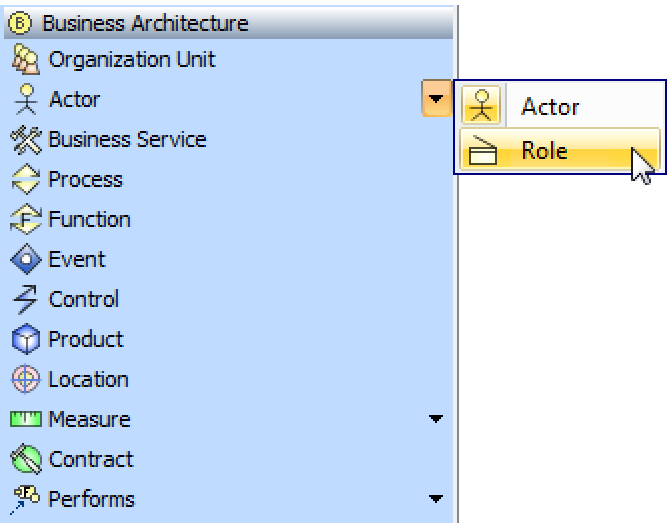
Sample


[Business Architecture]
«Actor» Client
«Actor» Client Relationship Manager
«ParticipatesIn»
«ParticipatesIn»
«Orchestrates»
«Orchestrates»
«Orchestrates»
«Business Service» Claim Payment Service
«Business Service» Customer Information Service
«Business Service» Claim Registration Service
«Process» Claim Damage

Related procedures

Creating TOGAF diagram

Related elements



Organization Unit
Actor
Role
Business Service
Process
Function
Event
Control
Product
Location
Measure
Service Quality
Contract

Available relations

Source element	Target element	Relation
Organization Unit	Organization Unit	Decomposes
Organization Unit	Function	Owns
Organization Unit	Location	Operates In
Organization Unit	Product	Produces
Actor	Actor	Decomposes
Actor	Function	Performs
Actor	Function	Interacts With
Actor	Business Service	Consumes
Actor	Event	Generates
Actor	Event	Resolves
Actor	Location	Operates In
Actor	Organization Unit	Belongs To
Actor	Process	Participates In
Actor	Role	Performs Task In
Role	Role	Decomposes
Role	Function	Accesses
Business Service	Business Service	Decomposes
Business Service	Event	Resolves
Business Service	Data Entity	Consumes
Business Service	Function	Provides Governed Interface To Access
Business Service	Process	Supports

Source element	Target element	Relation
Business Service	Service Quality	Meets
Process	Process	Decomposes
Process	Event	Generates
Process	Event	Resolves
Process	Function	Orchestrates
Process	Product	Produces
Process	Process	Precedes/Follows
Function	Function	Decomposes
Function	Function	Communicates With
Function	Actor	Supports
Control	Process	Ensures Correct Operation Of
Location	Location	Decomposes
Measure	Measure	Decomposes
Measure	Business Service	Sets Performance Criteria For
Contract	Business Service	Governs and Measures
Contract	Service Quality	Meets

Related diagrams

Architecture Vision diagram
Applications Architecture diagram
Requirements Management diagram

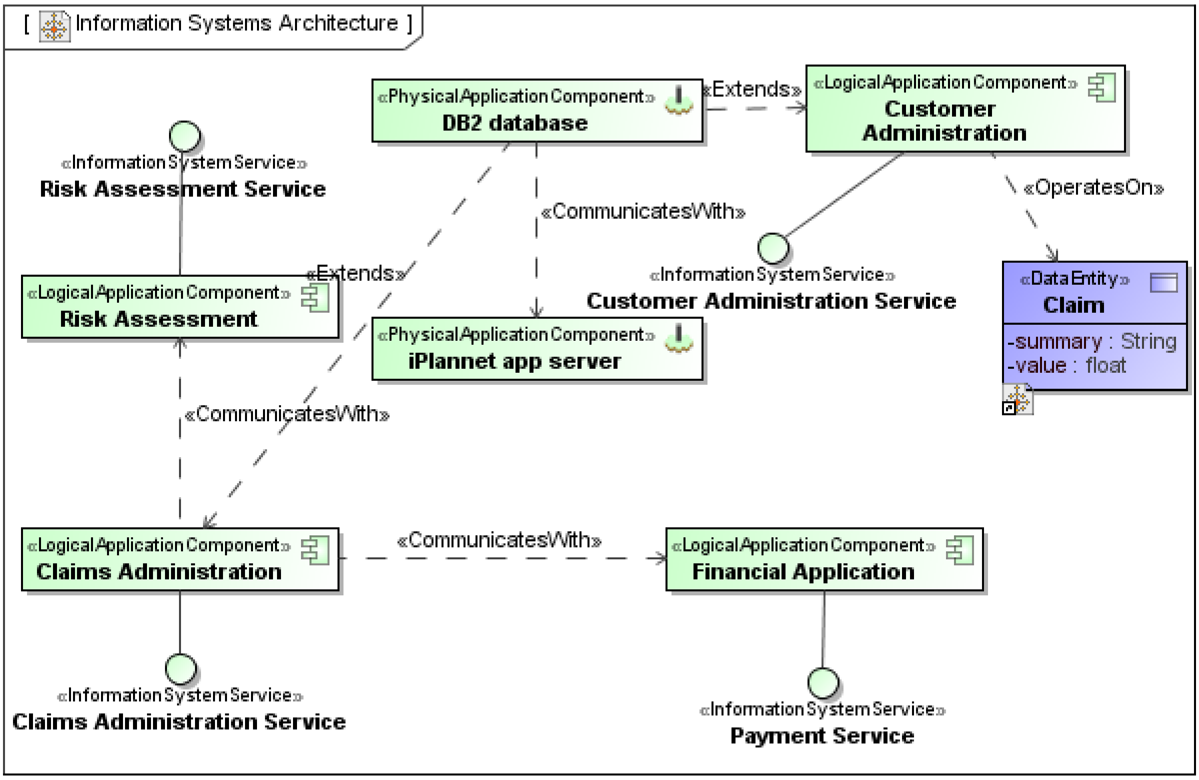
2.4 Applications Architecture diagram

Description

The goal of Application Architecture is to define what kinds of application systems are relevant to the enterprise, and what those applications need to do in order to manage data and to present information to the human and computer actors in the enterprise.

The applications are not described as computer systems, but as logical groups of capabilities that manage the data objects in the Data Architecture and support the business functions in the Business Architecture. The applications and their capabilities are defined without reference to particular technologies

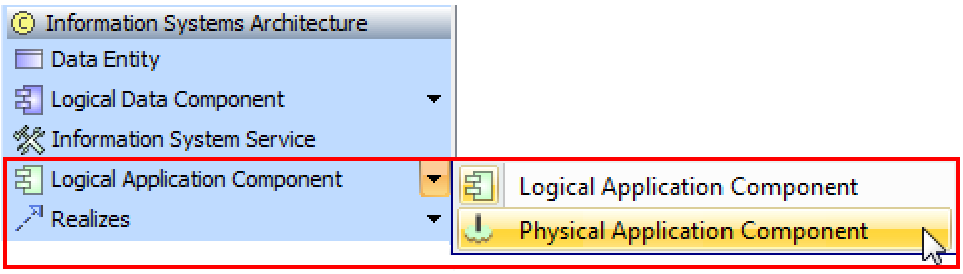
Sample


[Information Systems Architecture]
«PhysicalApplicationComponent»
DB2 database
«Extends»
«LogicalApplicationComponent»
Customer Administration
«InformationSystemService»
Risk Assessment Service
«CommunicatesWith»
«OperatesOn»
«Extends»
«LogicalApplicationComponent»
Risk Assessment
«InformationSystemService»
Customer Administration Service
«PhysicalApplicationComponent»
iPlannet app server
«DataEntity»
Claim
-summary : String
-value : float
«CommunicatesWith»
«LogicalApplicationComponent»
Claims Administration
«CommunicatesWith»
«LogicalApplicationComponent»
Financial Application
«InformationSystemService»
Claims Administration Service
«InformationSystemService»
Payment Service

Related procedures

Creating TOGAF diagram

Related elements


Information Systems Architecture
Data Entity
Logical Data Component
Information System Service
Logical Application Component
Realizes
Logical Application Component
Physical Application Component

Information System Service

Logical Application Component

Physical Application Component

Available relations

Source element	Target element	Relation
Information System Service	Information System Service	Decomposes
Information System Service	Event	Resolves
Information System Service	Data Entity	Consumes
Information System Service	Function	Provides Governed Interface To Access
Information System Service	Process	Supports
Information System Service	Service Quality	Meets
Logical Application Component	Logical Application Component	Decomposes
Logical Application Component	Logical Application Component	Communicates With
Logical Application Component	Business Service	Implements
Logical Application Component	Data Entity	Operates On
Physical Application Component	Physical Application Component	Decomposes
Physical Application Component	Physical Application Component	Communicates With
Physical Application Component	Location	Is Hosted In
Physical Application Component	Logical Application Component	Extends
Physical Application Component	Physical Data Component	Encapsulates

Related diagrams

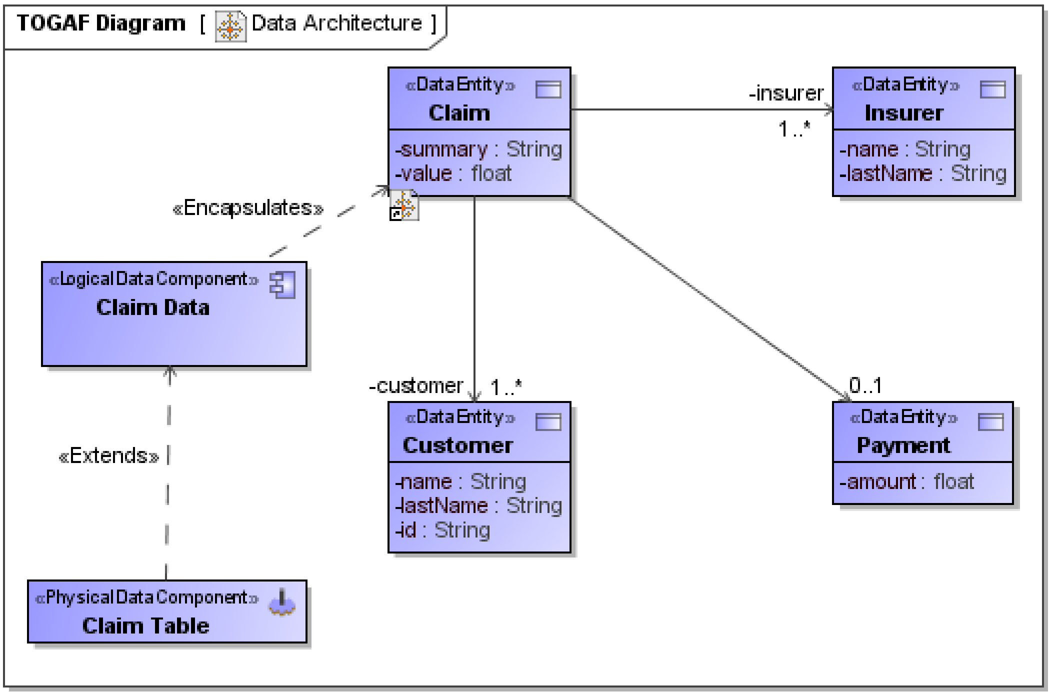
- Business Architecture diagram
- Technology Architecture diagram
- Requirements Management diagram
- Data Architecture diagram

2.5 Data Architecture diagram

Description

The goal of Data Architecture is to define the data entities relevant to the enterprise, not to design logical or physical storage systems. (However, linkages to existing files and databases may be developed, and may demonstrate significant areas for improvement.).

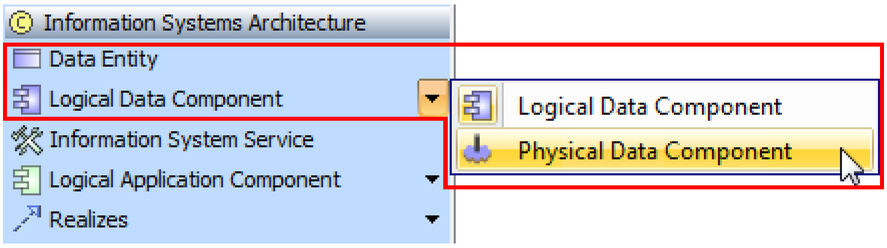
Sample



Related procedures

Creating TOGAF diagram

Related elements


Information Systems Architecture
Data Entity
Logical Data Component
Information System Service
Logical Application Component
Realizes
Logical Data Component
Physical Data Component

Data Entity

Logical Data Component

Physical Data Component

Available relations

Source element	Target element	Relation
Data Entity	Data Entity	Decomposes
Data Entity	Data Entity	Relates To
Data Entity	Business Service	Is Accessed And Updated Through
Logical Data Component	Data Entity	Encapsulates
Physical Data Component	Physical Data Component	Decomposes
Physical Data Component	Location	Is Hosted In
Physical Data Component	Logical Data Component	Extends
Physical Data Component	Physical Application Component	Encapsulates

Related diagrams

Applications Architecture diagram
Business Architecture diagram
Technology Architecture diagram
Requirements Management diagram

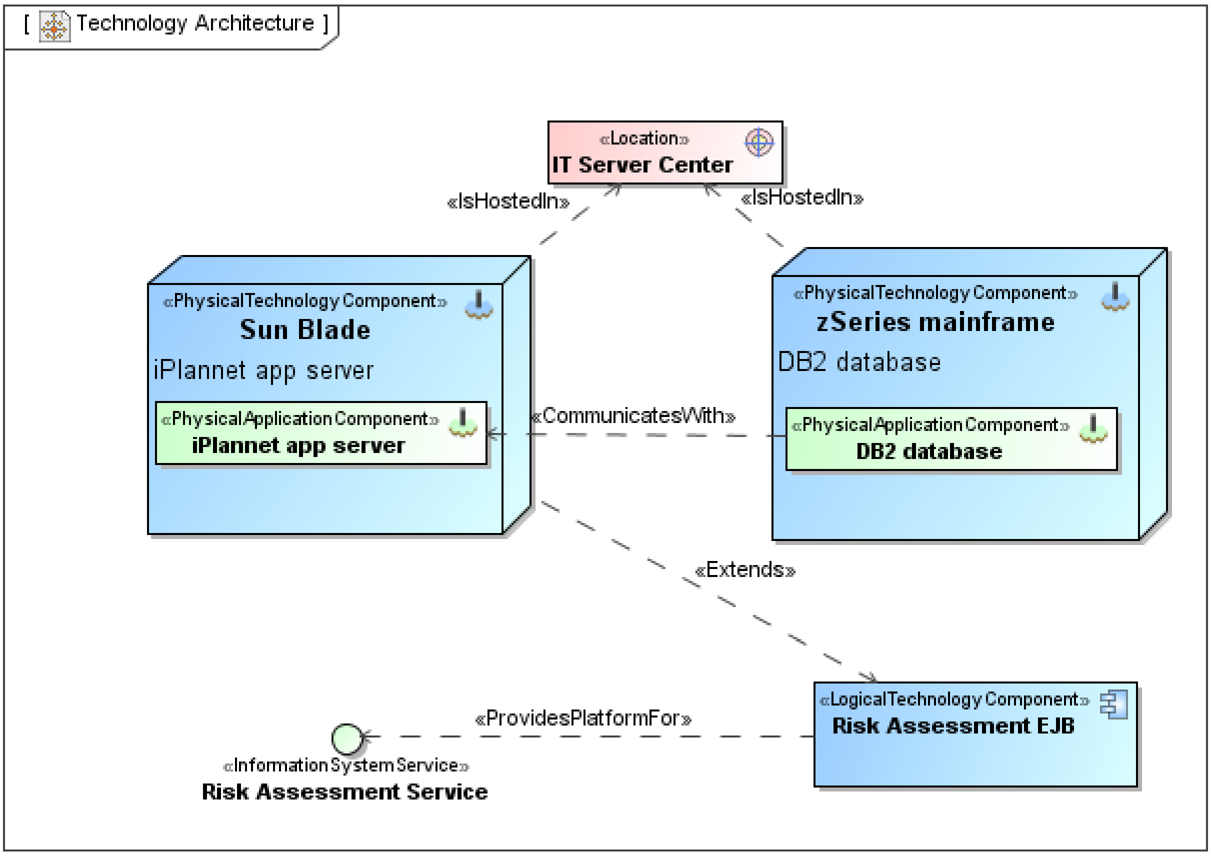
2.6 Technology Architecture diagram

Description

The Technology Architecture phase seeks to map application components defined in the Application Architecture phase into a set of technology components, which represent software and hardware components, available from the market or configured within the organization into technology platforms.

As Technology Architecture defines the physical realization of an architectural solution, it has strong links to implementation and migration planning. Technology Architecture completes the set of architectural information and therefore supports cost assessment for particular migration scenarios.

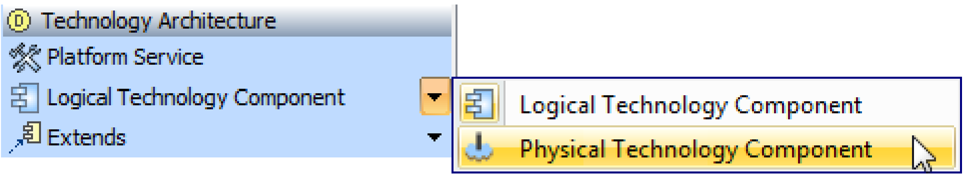
Sample



Related procedures

[Creating TOGAF diagram]()

Related elements


Technology Architecture
Platform Service
Logical Technology Component
Extends
Logical Technology Component
Physical Technology Component

[Platform Service]()
[Logical Technology Component]()
[Physical Technology Component]()

Available relations

Source element	Target element	Relation
Platform Service	Platform Service	Decomposes
Platform Service	Event	Resolves
Platform Service	Data Entity	Consumes
Platform Service	Function	Provides Governed Interface To Access
Platform Service	Service Quality	Meets
Logical Technology Component	Logical Technology Component	Decomposes
Logical Technology Component	Logical Technology Component	Is Dependent On
Logical Technology Component	Platform Service	Supplies
Logical Technology Component	Business Service	Provides Platform For
Physical Technology Component	Physical Technology Component	Decomposes
Physical Technology Component	Physical Technology Component	Is Dependent On
Physical Technology Component	Location	Is Hosted In
Physical Technology Component	Physical Application Component	Realizes
Physical Technology Component	Logical Technology Component	Extends

Related diagrams

Applications Architecture diagram
Data Architecture diagram
Requirements Management diagram

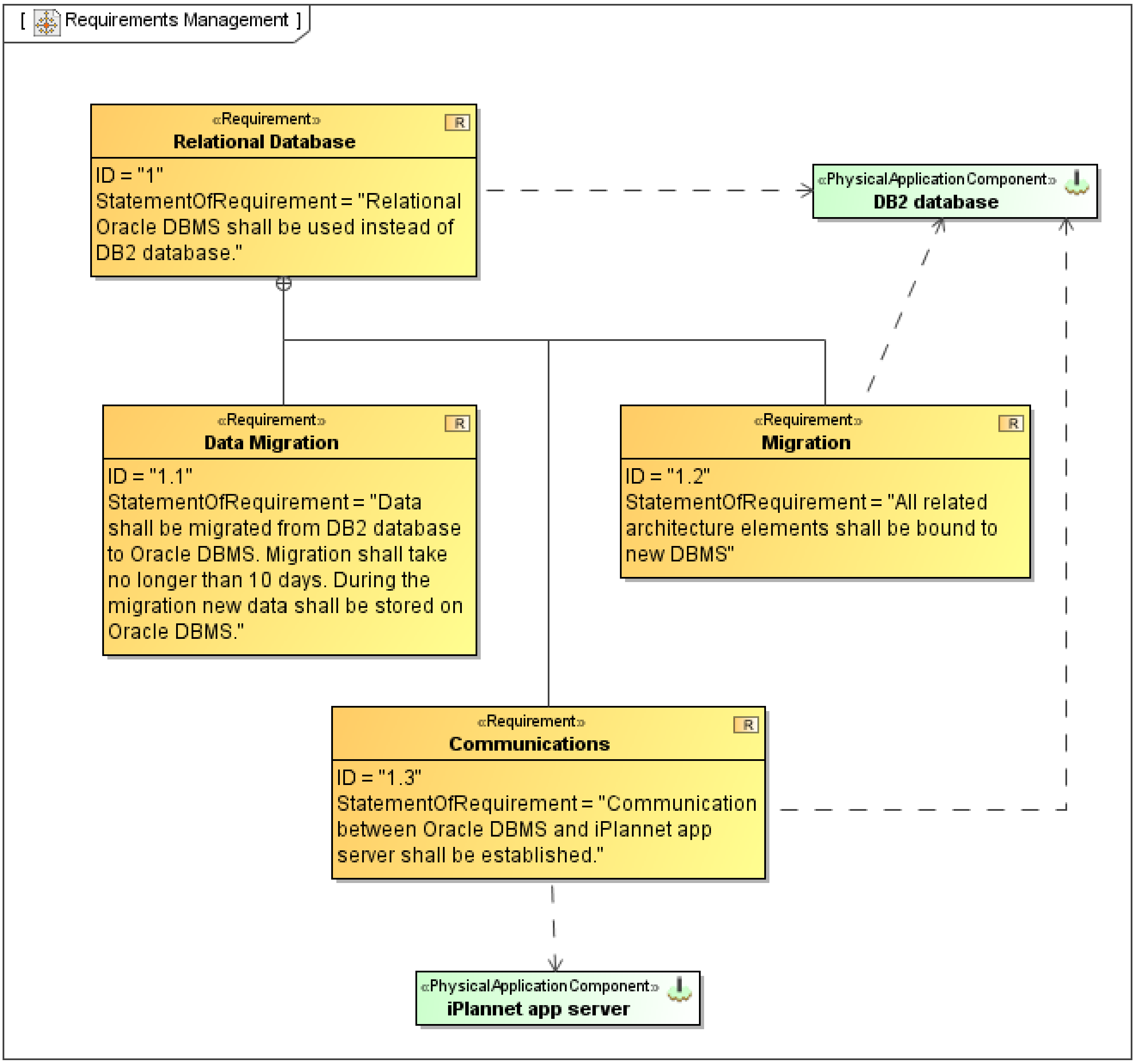
2.7 Requirements Management diagram

Description

The objective of the Requirements Management phase is to define a process whereby requirements for enterprise architecture are identified, stored, and fed into and out of the relevant ADM phases

Requirements Management circle denotes, not a static set of requirements, but a dynamic process whereby requirements for enterprise architecture and subsequent changes to those requirements are identified, stored, and fed into and out of the relevant Architecture Development Method (ADM) phases. Architecture often deals with drivers and constraints, many of which by their very nature are beyond the control of the enterprise (changing market conditions, new legislation, etc.), and which can produce changes in requirements in an unforeseen manner.

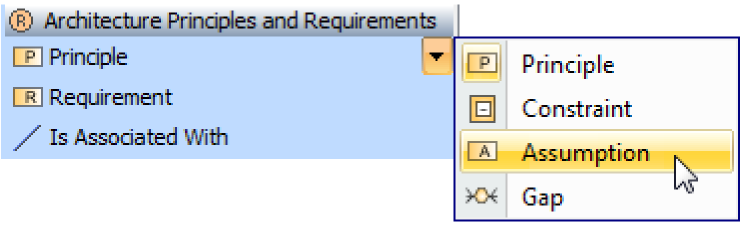
Sample


[Requirements Management]
«Requirement»
Relational Database
R
ID = "1"
StatementOfRequirement = "Relational Oracle DBMS shall be used instead of DB2 database."
«PhysicalApplicationComponent»
DB2 database
«Requirement»
Data Migration
R
ID = "1.1"
StatementOfRequirement = "Data shall be migrated from DB2 database to Oracle DBMS. Migration shall take no longer than 10 days. During the migration new data shall be stored on Oracle DBMS."
«Requirement»
Migration
R
ID = "1.2"
StatementOfRequirement = "All related architecture elements shall be bound to new DBMS"
«Requirement»
Communications
R
ID = "1.3"
StatementOfRequirement = "Communication between Oracle DBMS and iPlannet app server shall be established."
«PhysicalApplicationComponent»
iPlannet app server

Related procedures

Creating TOGAF diagram

Related elements



- Principle
- Constraint
- Assumption
- Gap
- Requirement

Available relations

Source element	Target element	Relation
Principle	Any element	Is Associated With
Constraint	Any element	Is Associated With
Assumption	Any element	Is Associated With
Gap	Any element	Is Associated With
Requirement	Any element	Is Associated With

Related diagrams

- Architecture Vision diagram
- Business Architecture diagram
- Applications Architecture diagram
- Technology Architecture diagram

3 TOGAF ELEMENTS

Learn about TOGAF elements (metamodel objects) in the following sections:

- All Elements
- Actor
- Application Component
- Assumption
- Business Service
- Capability
- Constraint
- Contract
- Control
- Data Entity
- Driver
- Event
- Function
- Gap
- Goal
- Information System Service
- Location
- Logical Application Component
- Logical Data Component
- Logical Technology Component
- Measure
- Objective
- Organization Unit
- Physical Application Component
- Physical Data Component
- Physical Technology Component
- Platform Service
- Principle
- Process
- Product
- Requirement
- Role
- Service
- Service Quality
- Technology Component
- Work Package

Related web resources

Content Metamodel Objects. In *TOGAF 9*. Retrieved April 18, 2011, from http://pubs.opengroup.org/architecture/togaf9-doc/arch/chap34.html#tag_34_05

Content Metamodel Attributes. In *TOGAF 9*. Retrieved April 18, 2011, from http://pubs.opengroup.org/architecture/togaf9-doc/arch/chap34.html#tag_34_06

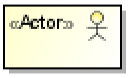
3.1 All Elements

Generic properties

Property	Description
ID	Unique identifier for the element.
Name	Brief name of the architecture object.
Description	Textual description of the architecture object.
Category	User-definable categorization taxonomy for each element.
Source	Location from where the information was collected.
Owner	Owner of the element.

3.2 Actor

Notation



Description

A person, organization, or system that has a role that initiates or interacts with activities; for example, a sales representative who travels to visit customers. Actors may be internal or external to an organization. In the automotive industry, an original equipment manufacturer would be considered an actor by an automotive dealership that interacts with its supply chain activities.

Element-specific properties

Property	Description
# FTEs	Estimated number of FTEs that operate as this Actor.
Actor goal	Objectives that this actor has, in general terms.
Actor tasks	Tasks that this actor performs, in general terms.

Extension

UML Class

Related diagrams

Business Architecture diagram

3.3 Application Component

Notation

N/A

Description

An encapsulation of application functionality aligned to implementation structure. For example, a purchase request processing application.

Element-specific properties

Property	Description
Standards class	Non-Standard, Proposed Standard, Provisional Standard, Standard, Phasing-Out Standard, Retired Standard.
Standard creation date	If the product is a standard, when the standard was created.
Last standard review date	Last date that the standard was reviewed.
Next standard review date	Next date for the standard to be reviewed.
Retire date	Date when the standard was/will be retired.

Extension

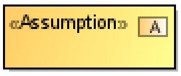
UML Component

Related elements

- Logical Application Component
- Physical Application Component

3.4 Assumption

Notation



Description

A statement of probable fact that has not been fully validated at this stage, due to external constraints. For example, it may be assumed that an existing application will support a certain set of functional requirements, although those requirements may not yet have been individually validated.

Element-specific properties

N/A

Extension

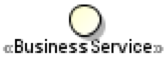
UML Class

Related diagrams

Requirements Management diagram

3.5 Business Service

Notation



Description

Supports business capabilities through an explicitly defined interface and is explicitly governed by an organization.

Element-specific properties

Property	Description
Standards class	Non-Standard, Proposed Standard, Provisional Standard, Standard, Phasing-Out Standard, Retired Standard.
Standard creation date	If the product is a standard, when the standard was created.
Last standard review date	Last date that the standard was reviewed.
Next standard review date	Next date for the standard to be reviewed.
Retire date	Date when the standard was/will be retired.

Extension

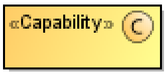
UML Interface

Related diagrams

Business Architecture diagram

3.6 Capability

Notation



Description

A business-focused outcome that is delivered by the completion of one or more work packages. Using a capability-based planning approach, change activities can be sequenced and grouped in order to provide continuous and incremental business value.

Element-specific properties

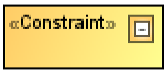
Property	Description
Business value	Describes how this capability provides value to the enterprise.
Increments	Lists possible maturity/quality levels for the capability.

Extension

UML Class

3.7 Constraint

Notation



Description

An external factor that prevents an organization from pursuing particular approaches to meet its goals. For example, customer data is not harmonized within the organization, regionally or nationally, constraining the organization's ability to offer effective customer service.

Element-specific properties

N/A

Extension

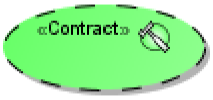
UML Class

Related diagrams

Requirements Management diagram

3.8 Contract

Notation



Description

An agreement between a service consumer and a service provider that establishes functional and non-functional parameters for interaction.

Element-specific properties

Property	Description
Behavior characteristics	Functional behavior to be supported within the scope of the contract.
Service name “caller”	Consuming service.
Service name “called”	Providing service.
Service quality characteristics	Non-functional behavior to be supported within the scope of the contract.
Availability characteristics	Degree to which something is available for use.
Service times	Hours during which the service must be available.
Manageability characteristics	Ability to gather information about the state of something and control it.
Serviceability characteristics	Ability to identify problems and take corrective action, such as to repair or upgrade a component in a running system.
Performance characteristics	Ability of a component to perform its tasks in an appropriate time.
Response requirements	Response times that the service provider must meet for particular operations.
Reliability characteristics	Resistance to failure.
Quality of information required	Contracted requirements on accuracy and completeness of information.
Contract control requirements	Level of governance and enforcement applied to the contractual parameters for overall service.
Result control requirements	Measures in place to ensure that each service request meets contracted criteria.
Recoverability characteristics	Ability to restore a system to a working state after an interruption.
Locatability characteristics	Ability of a system to be found when needed.
Security characteristics	Ability of a system to prevent unauthorized access to functions and data.
Privacy characteristics	Protection of data from unauthorized access.
Integrity characteristics	Ability of a system to ensure that data has not been corrupted.
Credibility characteristics	Ability of a system to ensure that the service request originates from an authorized source.
Localization characteristics	Ability of a service to support localized variants for different consumer groups.
Internationalization characteristics	Ability of a service to support international variations in business logic and data representation (such as character set).

Property	Description
Interoperability characteristics	Ability of the service to interoperate with different technical environments, inside and outside of the organization.
Scalability characteristics	Ability of the service to grow or shrink its performance or capacity appropriately to the demands of the environment in which it operates.
Portability characteristics	Of data, people, applications, and components.
Extensibility characteristics	Ability to accept new functionality.
Capacity characteristics	Contracted capacity of the service provider to meet requests.
Throughput	Required throughput capacity.
Throughput period	Time period needed to deliver throughput capacity.
Growth	Expected future growth rate of service request.
Growth period	Time period needed to reach the expected growth rate.
Peak profile short term	Short-term profile of peak service traffic.
Peak profile long term	Long-term profile of peak service traffic.

Extension

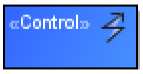
UML Collaboration

Related diagrams

Business Architecture diagram

3.9 Control

Notation



Description

A decision-making step with accompanying decision logic used to determine execution approach for a process or to ensure that a process complies with governance criteria. For example, a sign-off control on the purchase request processing process that checks whether the total value of the request is within the sign-off limits of the requester, or whether it needs escalating to higher authority.

Element-specific properties

N/A

Extension

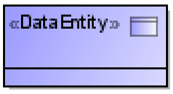
UML Class

Related diagrams

Business Architecture diagram

3.10 Data Entity

Notation



Description

An encapsulation of data that is recognized by a business domain expert as a thing. Logical data entities can be tied to applications, repositories, and services and may be structured according to implementation considerations.

Element-specific properties

Property	Description
Category	The following categories of data entity apply: Message, Internally Stored Entity.
Privacy classification	Level of restriction placed on access to the data.
Retention classification	Level of retention to be placed on the data.

Extension

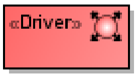
UML Class

Related diagrams

Data Architecture diagram

3.11 Driver

Notation



Description

An external or internal condition that motivates the organization to define its goals. An example of an external driver is a change in regulation or compliance rules which, for example, require changes to the way an organization operates; i.e., Sarbanes-Oxley in the US.

Element-specific properties

N/A

Extension

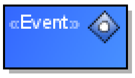
UML Class

Related diagrams

Architecture Vision diagram

3.12 Event

Notation



Description

An organizational state change that triggers processing events; may originate from inside or outside the organization and may be resolved inside or outside the organization.

Element-specific properties

N/A

Extension

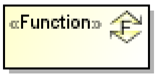
UML Class

Related diagrams

Business Architecture diagram

3.13 Function

Notation



Description

Delivers business capabilities closely aligned to an organization, but not necessarily explicitly governed by the organization. Also referred to as “business function”.

Element-specific properties

Property	Description
Standards class	Non-Standard, Proposed Standard, Provisional Standard, Standard, Phasing-Out Standard, Retired Standard.
Standard creation date	If the product is a standard, when the standard was created.
Last standard review date	Last date that the standard was reviewed.
Next standard review date	Next date for the standard to be reviewed.
Retire date	Date when the standard was/will be retired.

Extension

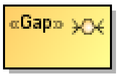
UML Activity

Related diagrams

Business Architecture diagram

3.14 Gap

Notation



Description

A statement of difference between two states. Used in the context of gap analysis, where the difference between the Baseline and Target Architecture is identified.

Element-specific properties

N/A

Extension

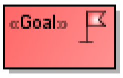
UML Class

Related diagrams

Requirements Management diagram

3.15 Goal

Notation



Description

A high-level statement of intent or direction for an organization. Typically used to measure success of an organization.

Element-specific properties

N/A

Extension

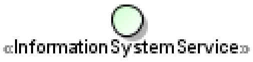
UML Class

Related diagrams

Architecture Vision diagram

3.16 Information System Service

Notation



Description

The automated elements of a business service. An information system service may deliver or support part or all of one or more business services.

Element-specific properties

Property	Description
Standards class	Non-Standard, Proposed Standard, Provisional Standard, Standard, Phasing-Out Standard, Retired Standard.
Standard creation date	If the product is a standard, when the standard was created.
Last standard review date	Last date that the standard was reviewed.
Next standard review date	Next date for the standard to be reviewed.
Retire date	Date when the standard was/will be retired.

Extension

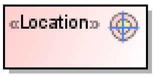
UML Interface

Related diagrams

Applications Architecture diagram

3.17 Location

Notation



Description

A place where business activity takes place and can be hierarchically decomposed.

Element-specific properties

Property	Description
Category	The following categories of Location apply: Region (applies to a grouping of countries or territory; e.g., South East Asia, UK, and Ireland), Country (applies to a single country; e.g., US), and Building (applies to a site of operation; where several offices are collected in a single city, this category may represent a city), Specific Location (applies to any specific location within a building, such as a server room).

Extension

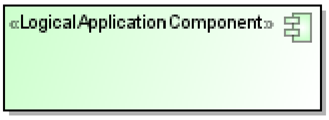
UML Data Type

Related diagrams

Business Architecture diagram

3.18 Logical Application Component

Notation



Description

An encapsulation of application functionality that is independent of a particular implementation. For example, the classification of all purchase request processing applications implemented in an enterprise.

Element-specific properties

Property	Description
Standards class	Non-Standard, Proposed Standard, Provisional Standard, Standard, Phasing-Out Standard, Retired Standard.
Standard creation date	If the product is a standard, when the standard was created.
Last standard review date	Last date that the standard was reviewed.
Next standard review date	Next date for the standard to be reviewed.
Retire date	Date when the standard was/will be retired.

Extension

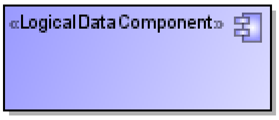
UML Component

Related diagrams

Applications Architecture diagram

3.19 Logical Data Component

Notation



Description

A boundary zone that encapsulates related data entities to form a logical location to be held; for example, external procurement information.

Element-specific properties

Property	Description
Standards class	Non-Standard, Proposed Standard, Provisional Standard, Standard, Phasing-Out Standard, Retired Standard.
Standard creation date	If the product is a standard, when the standard was created.
Last standard review date	Last date that the standard was reviewed.
Next standard review date	Next date for the standard to be reviewed.
Retire date	Date when the standard was/will be retired.

Extension

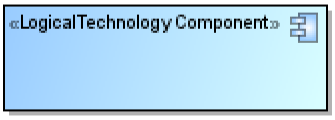
UML Component

Related diagrams

Data Architecture diagram

3.20 Logical Technology Component

Notation



Description

An encapsulation of technology infrastructure that is independent of a particular product. A class of technology product; for example, supply chain management software as part of an Enterprise Resource Planning (ERP) suite, or a Commercial Off-The-Shelf (COTS) purchase request processing enterprise service.

Element-specific properties

Property	Description
Standards class	Non-Standard, Proposed Standard, Provisional Standard, Standard, Phasing-Out Standard, Retired Standard.
Standard creation date	If the product is a standard, when the standard was created.
Last standard review date	Last date that the standard was reviewed.
Next standard review date	Next date for the standard to be reviewed.
Retire date	Date when the standard was/will be retired.
Category	Logical Technology Components are categorized according to the TOGAF TRM, which may be extended to meet the needs of an individual organization.

Extension

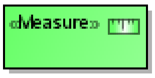
UML Component

Related diagrams

Technology Architecture diagram

3.21 Measure

Notation



Description

An indicator or factor that can be tracked, usually on an ongoing basis, to determine success or alignment with objectives and goals.

Element-specific properties

N/A

Extension

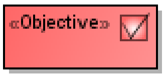
UML Data Type

Related diagrams

Business Architecture diagram

3.22 Objective

Notation



Description

A time-bounded milestone for an organization used to demonstrate progress towards a goal; for example, “Increase capacity utilization by 30% by the end of 2009 to support the planned increase in market share”.

Element-specific properties

N/A

Extension

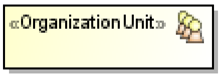
UML Class

Related diagrams

Architecture Vision diagram

3.23 Organization Unit

Notation



Description

A self-contained unit of resources with line management responsibility, goals, objectives, and measures. Organizations may include external parties and business partner organizations.

Element-specific properties

Property	Description
Headcount	Number of FTEs working within the organization.

Extension

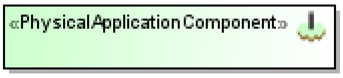
UML Class

Related diagrams

Business Architecture diagram

3.24 Physical Application Component

Notation



Description

An application, application module, application service, or other deployable component of functionality. For example, a configured and deployed instance of a Commercial Off-The-Shelf (COTS) Enterprise Resource Planning (ERP) supply chain management application.

Element-specific properties

Property	Description
Lifecycle status	Proposed, In Development, Live, Phasing Out, Retired.
Standards class	Non-Standard, Proposed Standard, Provisional Standard, Standard, Phasing-Out Standard, Retired Standard.
Standard creation date	If the product is a standard, when the standard was created.
Last standard review date	Last date that the standard was reviewed.
Next standard review date	Next date for the standard to be reviewed.
Retire date	Date when the standard was/will be retired.
Initial live date	Date when the first release of the application was/will be released into production.
Date of last release	Date when the last release of the application was released into production.
Date of next release	Date when the next release of the application will be released into production.
Retirement date	Date when the application was/will be retired.
Availability characteristics	Degree to which something is available for use.
Service times	Hours during which the application must be available.
Manageability characteristics	Ability to gather information about the state of something and control it.
Serviceability characteristics	Ability to identify problems and take corrective action, such as to repair or upgrade a component in a running system.
Performance characteristics	Ability of a component to perform its tasks in an appropriate time.
Reliability characteristics	Resistance to failure.
Recoverability characteristics	Ability to restore a system to a working state after an interruption.
Locatability characteristics	Ability of a system to be found when needed.
Security characteristics	Ability of a system to prevent unauthorized access to functions and data.
Privacy characteristics	Protection of data from unauthorized access.
Integrity characteristics	Ability of a system to ensure that data has not been corrupted.
Credibility characteristics	Ability of a system to ensure that the service request originates from an authorized source.
Localization characteristics	Ability of a service to support localized variants for different consumer groups.
Internationalization characteristics	Ability of a service to support international variations in business logic and data representation (such as character set).

Property	Description
Interoperability characteristics	Ability of the service to interoperate with different technical environments, inside and outside of the organization.
Scalability characteristics	Ability of the service to grow or shrink its performance or capacity appropriately to the demands of the environment in which it operates.
Portability characteristics	Of data, people, applications, and components.
Extensibility characteristics	Ability to accept new functionality.
Capacity characteristics	Contracted capacity of the service provider to meet requests.
Throughput	Required throughput capacity.
Throughput period	Time period needed to deliver throughput capacity.
Growth	Expected future growth rate of service request.
Growth period	Time period needed to reach the expected growth rate.
Peak profile short term	Short-term profile of peak service traffic.
Peak profile long term	Long-term profile of peak service traffic.

Extension

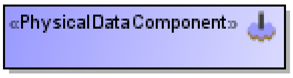
UML Artifact

Related diagrams

Applications Architecture diagram

3.25 Physical Data Component

Notation



Description

A boundary zone that encapsulates related data entities to form a physical location to be held. For example, a purchase order business object, comprising purchase order header and item business object nodes.

Element-specific properties

Property	Description
Standards class	Non-Standard, Proposed Standard, Provisional Standard, Standard, Phasing-Out Standard, Retired Standard.
Standard creation date	If the product is a standard, when the standard was created.
Last standard review date	Last date that the standard was reviewed.
Next standard review date	Next date for the standard to be reviewed.
Retire date	Date when the standard was/will be retired.

Extension

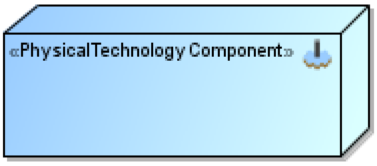
UML Artifact

Related diagrams

Data Architecture diagram

3.26 Physical Technology Component

Notation



Description

A specific technology infrastructure product or technology infrastructure product instance. For example, a particular product version of a Commercial Off-The-Shelf (COTS) solution, or a specific brand and version of server.

Element-specific properties

Property	Description
Standards class	Non-Standard, Proposed Standard, Provisional Standard, Standard, Phasing-Out Standard, Retired Standard.
Standard creation date	If the product is a standard, when the standard was created.
Last standard review date	Last date that the standard was reviewed.
Next standard review date	Next date for the standard to be reviewed.
Retire date	Date when the standard was/will be retired.
Category	Physical Technology Components are categorized according to the TOGAF TRM, which may be extended to meet the needs of an individual organization.
Product name	Name of the product making up the technology component.
Module name	Module, or other sub-product, name making up the technology component.
Vendor	Vendor providing the technology component.
Version	Version of the product making up the technology component.

Extension

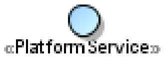
UML Node

Related diagrams

Technology Architecture diagram

3.27 Platform Service

Notation



Description

A technical capability required to provide enabling infrastructure that supports the delivery of applications.

Element-specific properties

Property	Description
Standards class	Non-Standard, Proposed Standard, Provisional Standard, Standard, Phasing-Out Standard, Retired Standard.
Category	Platform Services are categorized according to the TOGAF TRM, which may be extended to meet the needs of an individual organization.

Extension

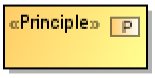
UML Interface

Related diagrams

[Technology Architecture diagram]

3.28 Principle

Notation



Description

A qualitative statement of intent that should be met by the architecture. Has at least a supporting rationale and a measure of importance.

Element-specific properties

Property	Description
Category	The following categories of principle apply: Guiding Principle, Business Principle, Data Principle, Application Principle, Integration Principle, Technology Principle.
Priority	Priority of this principle relative to other principles.
Statement of principle	Statement of what the principle is.
Rationale	Statement of why the principle is required and the outcome to be reached.
Implication	Statement of what the principle means in practical terms.
Metric	Identifies mechanisms that will be used to measure whether the principle has been met or not.

Extension

UML Class

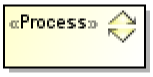
Related diagrams

[Preliminary diagram]

[Requirements Management diagram]

3.29 Process

Notation



Description

A process represents flow of control between or within functions and/or services (depends on the granularity of definition).

Processes represent a sequence of activities that together achieve a specified outcome, can be decomposed into sub-processes, and can show operation of a function or service (at next level of detail). Processes may also be used to link or compose organizations, functions, services, and processes.

Element-specific properties

Property	Description
Standards class	Non-Standard, Proposed Standard, Provisional Standard, Standard, Phasing-Out Standard, Retired Standard.
Standard creation date	If the product is a standard, when the standard was created.
Last standard review date	Last date that the standard was reviewed.
Next standard review date	Next date for the standard to be reviewed.
Retire date	Date when the standard was/will be retired.
Process criticality	Criticality of this process to business operations.Manual or automated
Manual or automated	Whether this process is supported by IT or is a manual process.
Process volumetrics	Data on frequency of process execution.

Extension

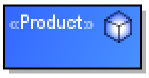
UML Activity

Related diagrams

Business Architecture diagram

3.30 Product

Notation



Description

Output generated by the business. The business product of the execution of a process.

Element-specific properties

N/A

Extension

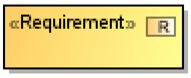
UML Class

Related diagrams

Business Architecture diagram

3.31 Requirement

Notation



Description

A quantitative statement of business need that must be met by a particular architecture or work package.

Element-specific properties

Property	Description
Statement of requirement	Statement of what the requirement is, including a definition of whether the requirement shall be met, should be met, or may be met.
Rationale	Statement of why the requirement exists.
Acceptance criteria	Statement of what tests will be carried out to ensure that the requirement will be met.

Extension

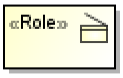
UML Class

Related diagrams

Requirements Management diagram

3.32 Role

Notation



Description

The usual or expected function of an actor, or the part somebody or something plays in a particular action or event. An actor may have a number of roles.

Element-specific properties

Property	Description
Estimated number of FTEs that operate in this Role	This metamodel object has only basic attributes.

Extension

UML Actor

Related diagrams

[Business Architecture diagram](#)

3.33 Service

Notation

N/A

Description

An element of behavior that provides specific functionality in response to requests from actors or other services. A service delivers or supports business capabilities, has an explicitly defined interface, and is explicitly governed. Services are defined for business, information systems, and platforms.

Element-specific properties

Property	Description
Standards class	Non-Standard, Proposed Standard, Provisional Standard, Standard, Phasing-Out Standard, Retired Standard.
Standard creation date	If the product is a standard, when the standard was created.
Last standard review date	Last date that the standard was reviewed.
Next standard review date	Next date for the standard to be reviewed.
Retire date	Date when the standard was/will be retired.

Extension

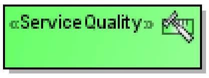
UML Component

Related elements

Business Service
Information System Service
Platform Service

3.34 Service Quality

Notation



Description

A preset configuration of non-functional attributes that may be assigned to a service or service contract.

Element-specific properties

N/A

Extension

UML Data Type

Related diagrams

Business Architecture diagram

3.35 Technology Component

Notation

N/A

Description

An encapsulation of technology infrastructure that represents a class of technology product or specific technology product.

Element-specific properties

Property	Description
Standards class	Non-Standard, Proposed Standard, Provisional Standard, Standard, Phasing-Out Standard, Retired Standard.

Extension

UML Component

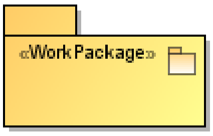
Related elements

Logical Technology Component

Physical Technology Component

3.36 Work Package

Notation



Description

A set of actions identified to achieve one or more objectives for the business. A work package can be a part of a project, a complete project, or a program.

Element-specific properties

Property	Description
Category	The following categories of work package apply: Work Package, Work Stream, Project, Program, Portfolio.
Capability delivered	Describes the contribution this work package makes to capability delivery.

Extension

UML Package

USING TOGAF PLUGIN

Learn about using the TOGAF plugin in the following sections:

Creating TOGAF diagram

All TOGAF diagrams are created using the same diagram.

You can create a TOGAF diagram in one of the following ways:

- Using a command from the shortcut menu of a package or element in the Containment tree.
- Using a command from the main menu.

To create a TOGAF diagram using a command from the shortcut menu of a package or an element:

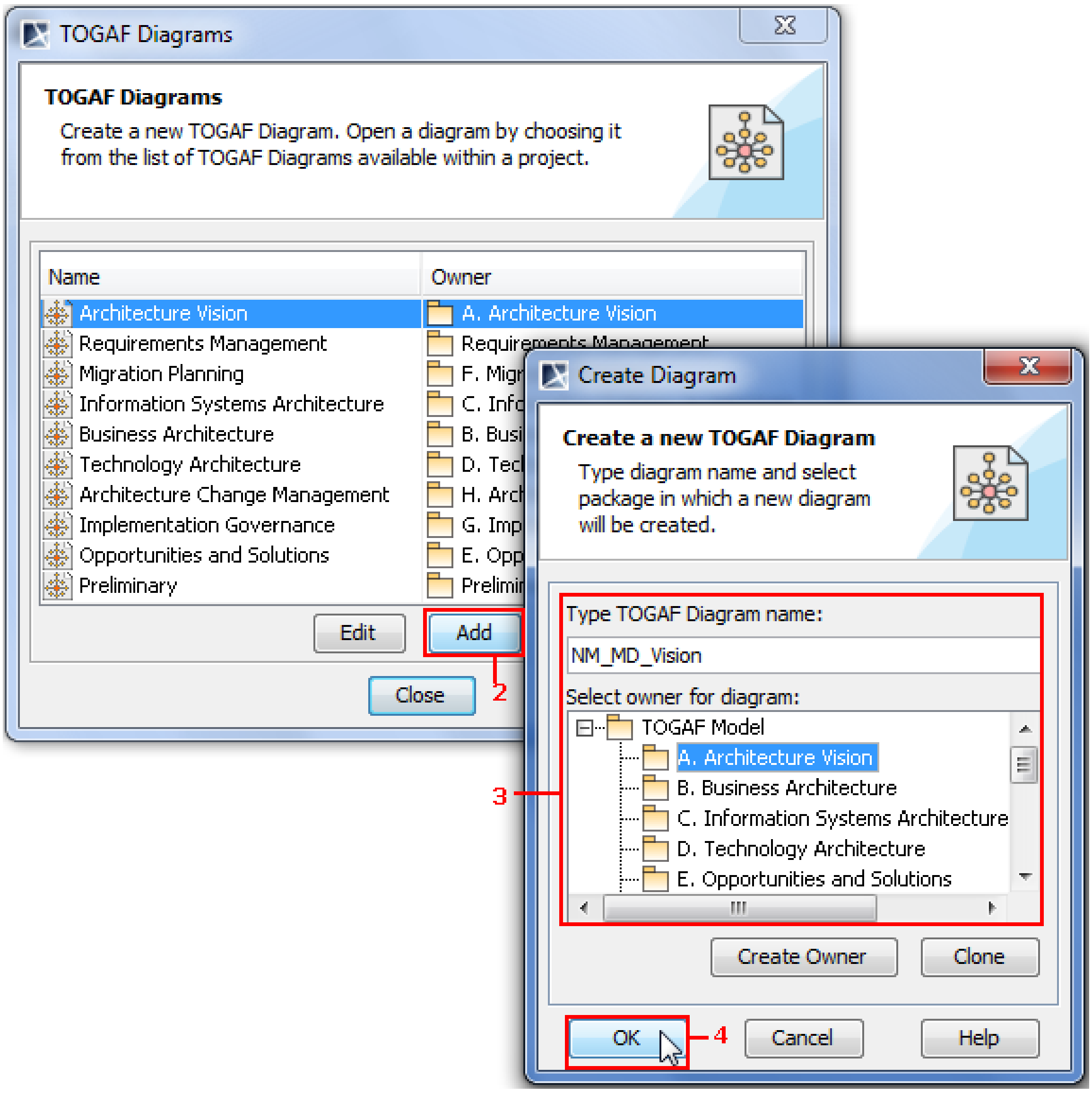
1. In the Containment tree, right-click either a package or an element wherein you want to create the TOGAF diagram.
2. Select **New Diagram** > **TOGAF diagram**.

To create a TOGAF diagram via the main menu:

1. From the **Diagrams** menu, select **Other Diagrams** > **TOGAF diagram**.
2. In the **TOGAF Diagrams** dialog click **Add**.
3. Specify the diagram name and select or create an owner for the diagram.

TIP!	You can select as a diagram owner a package that is already created for an appropriate phase in the TOGAF template.

4. Click **OK**.



TOGAF Diagrams

TOGAF Diagrams

Create a new TOGAF Diagram. Open a diagram by choosing it from the list of TOGAF Diagrams available within a project.

Name	Owner
Architecture Vision	A. Architecture Vision
Requirements Management	Requirements Management
Migration Planning	F. Migr
Information Systems Architecture	C. Info
Business Architecture	B. Busi
Technology Architecture	D. Tec
Architecture Change Management	H. Arch
Implementation Governance	G. Imp
Opportunities and Solutions	E. Opp
Preliminary	Prelimin

Create Diagram

Create a new TOGAF Diagram

Type diagram name and select package in which a new diagram will be created.

Type TOGAF Diagram name: NM_MD_Vision

Select owner for diagram:

- TOGAF Model
  - A. Architecture Vision
  - B. Business Architecture
  - C. Information Systems Architecture
  - D. Technology Architecture
  - E. Opportunities and Solutions

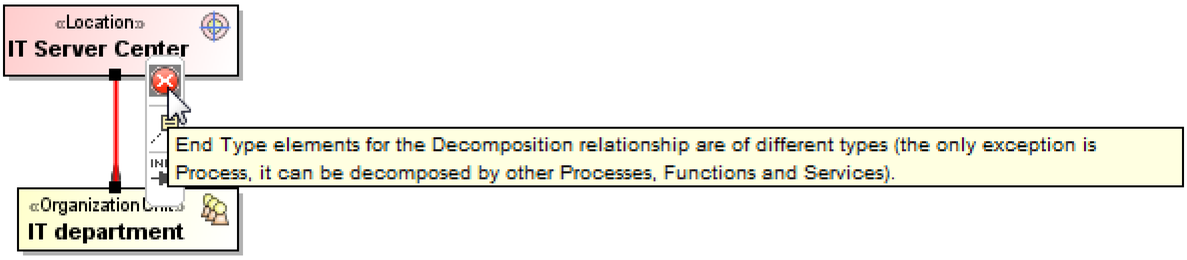
Figure 3 -- Creating new TOGAF diagram

Related diagrams

- [Preliminary diagram]
- [Architecture Vision diagram]
- [Business Architecture diagram]
- [Applications Architecture diagram]
- [Data Architecture diagram]
- [Technology Architecture diagram]
- [Requirements Management diagram]

Validating TOGAF Model

The active validation is by default enabled in a TOGAF model after it is created. All incorrect model elements and relations between them will be highlighted in red.



End Type elements for the Decomposition relationship are of different types (the only exception is Process, it can be decomposed by other Processes, Functions and Services).

Figure 4 -- Incorrect relation between two elements

You can disable active validation.

To disable the active validation:

- On the **Analyze** menu, point to **Validation** and then click to clear the **Enable Active Validation** check box.

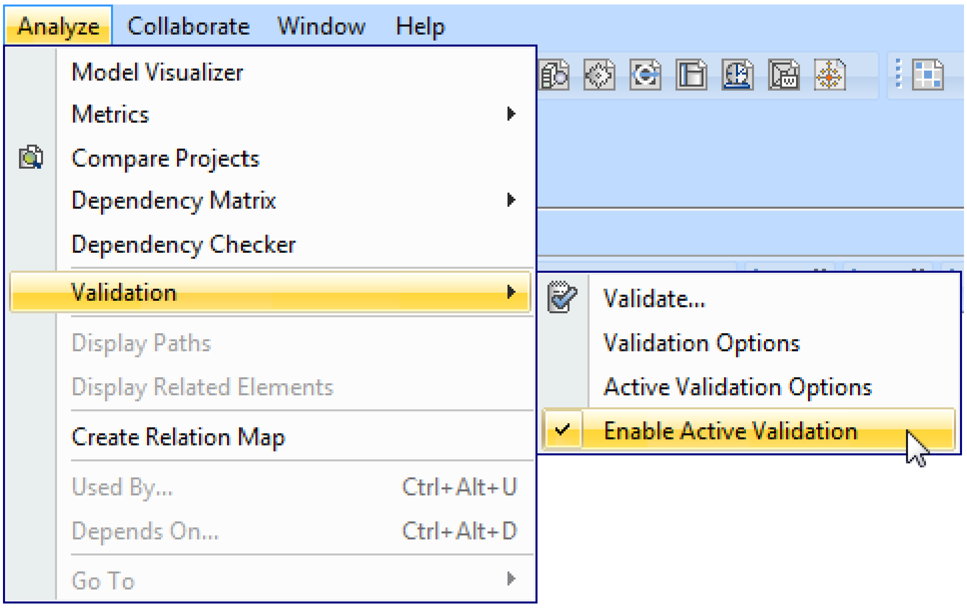


Figure 5 -- Disabling active validation

For more information about model validation see “Validation” in *MagicDraw UserManual.pdf*.

Using Relation Map

In order to analyze your TOGAF models you can use relation map, the feature introduced by MagicDraw.

Case study

Let us say we want to track relations between elements from different abstraction layers in one view. In this particular case we will find out what information system services and information system architecture components assist in realization of a business process named “Register claim”.

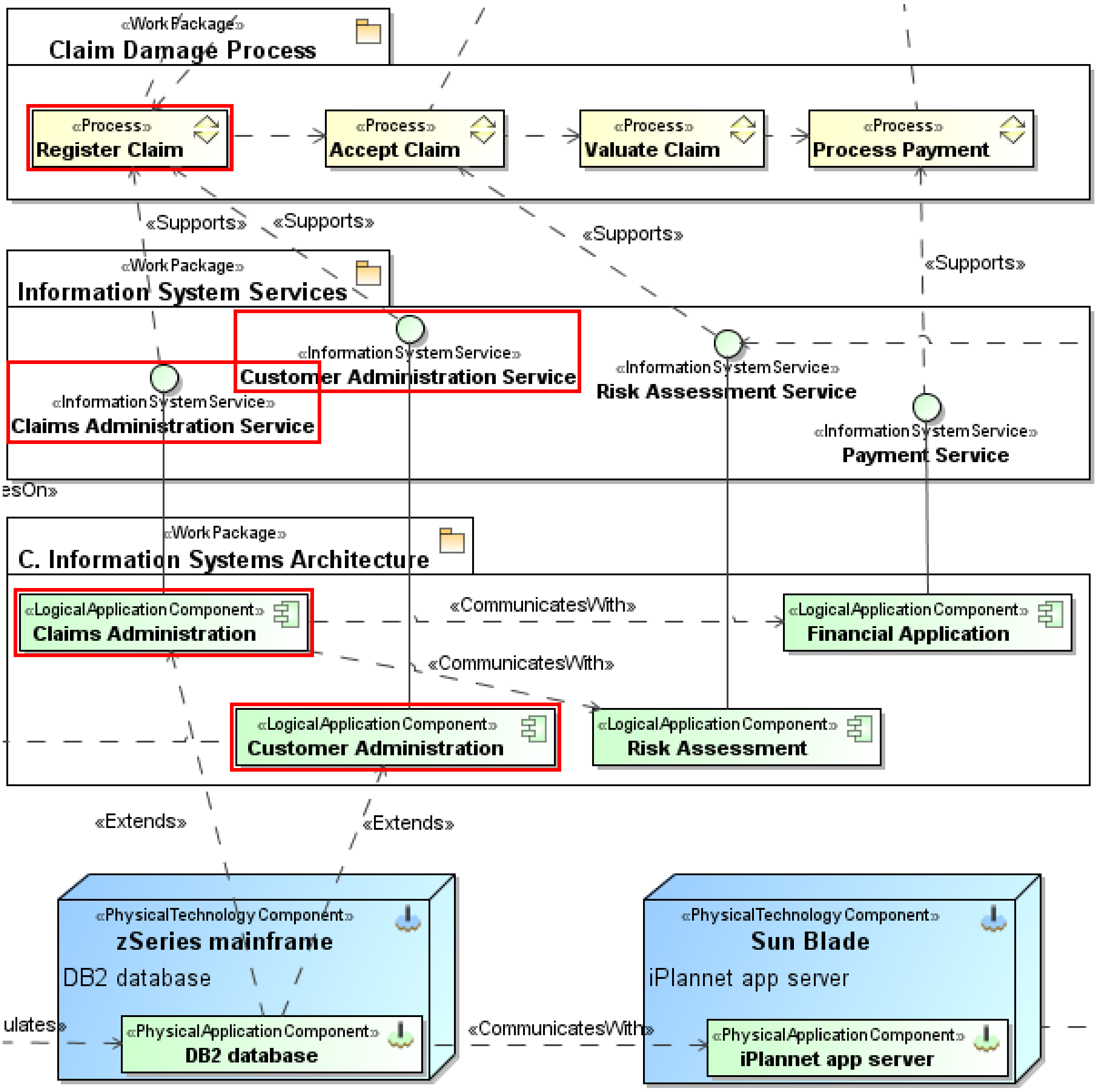

«Work Package»
Claim Damage Process
«Process»
Register Claim
«Process»
Accept Claim
«Process»
Valuate Claim
«Process»
Process Payment
«Supports»
«Supports»
«Supports»
«Supports»
«Work Package»
Information System Services
«Information System Service»
Customer Administration Service
«Information System Service»
Claims Administration Service
«Information System Service»
Risk Assessment Service
«Information System Service»
Payment Service
esOn»
«Work Package»
C. Information Systems Architecture
«Logical Application Component»
Claims Administration
«CommunicatesWith»
«Logical Application Component»
Financial Application
«CommunicatesWith»
«Logical Application Component»
Customer Administration
«Logical Application Component»
Risk Assessment
«Extends»
«Extends»
«Physical Technology Component»
zSeries mainframe
DB2 database
ulates»
«Physical Application Component»
DB2 database
«CommunicatesWith»
«Physical Technology Component»
Sun Blade
iPlannet app server
«Physical Application Component»
iPlannet app server

Figure 6 -- Elements assisting in realization of “Register Claim” business process

A relation map will be used to display relations between elements from the different layers of abstraction.

To display relations between the “Register Claim” business process, services, and components:

1. Create a relation map for the “Register claim” business process.
2. In the toolbar of the created relation map, click the **...** button next to the **Relation Criterion** box.
3. In the **Criterion Editor** dialog, select the Supports relation, as it was used to model dependencies between the business process and two information system services (see Figure 6).

4. Click **OK**. Elements that directly assist in realization of the business process will be displayed in the relation map.

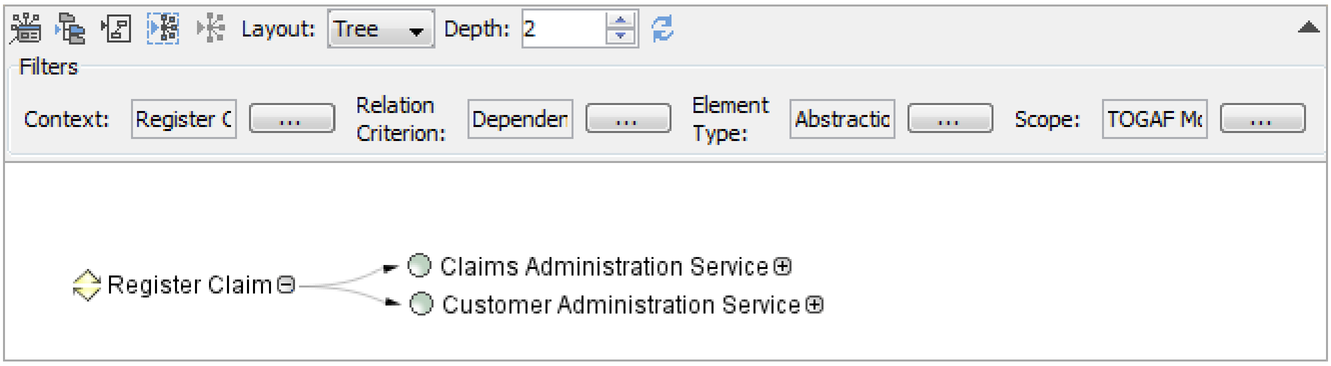


Figure 7 -- Displaying information system services that directly assist in realization of "Register claim" business process

5. Open the **Criterion Editor** dialog once again and select the Implements relation, as it was used to model dependencies between the information system services and the logical application components (see Figure 6).
6. Click **OK**. Elements that indirectly assist in realization of the business process will be displayed in the relation map.

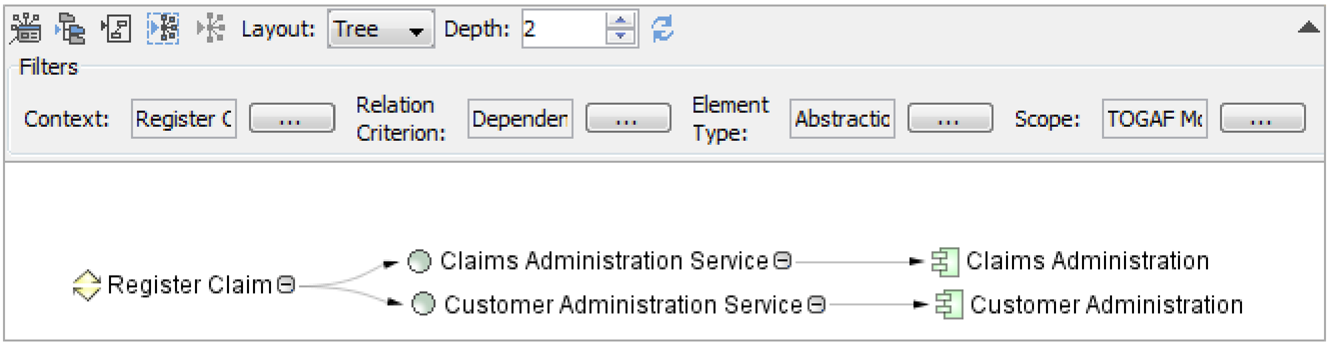


Figure 8 -- Displaying logical application components that indirectly assist in realization of "Register claim" business process

For more information about using relation maps, see "Relation Map" in *MagicDraw UserManual.pdf*.

Using Dependency Matrix

In order to analyze your TOGAF models you can use dependency matrix, the feature introduced by MagicDraw.

Case study

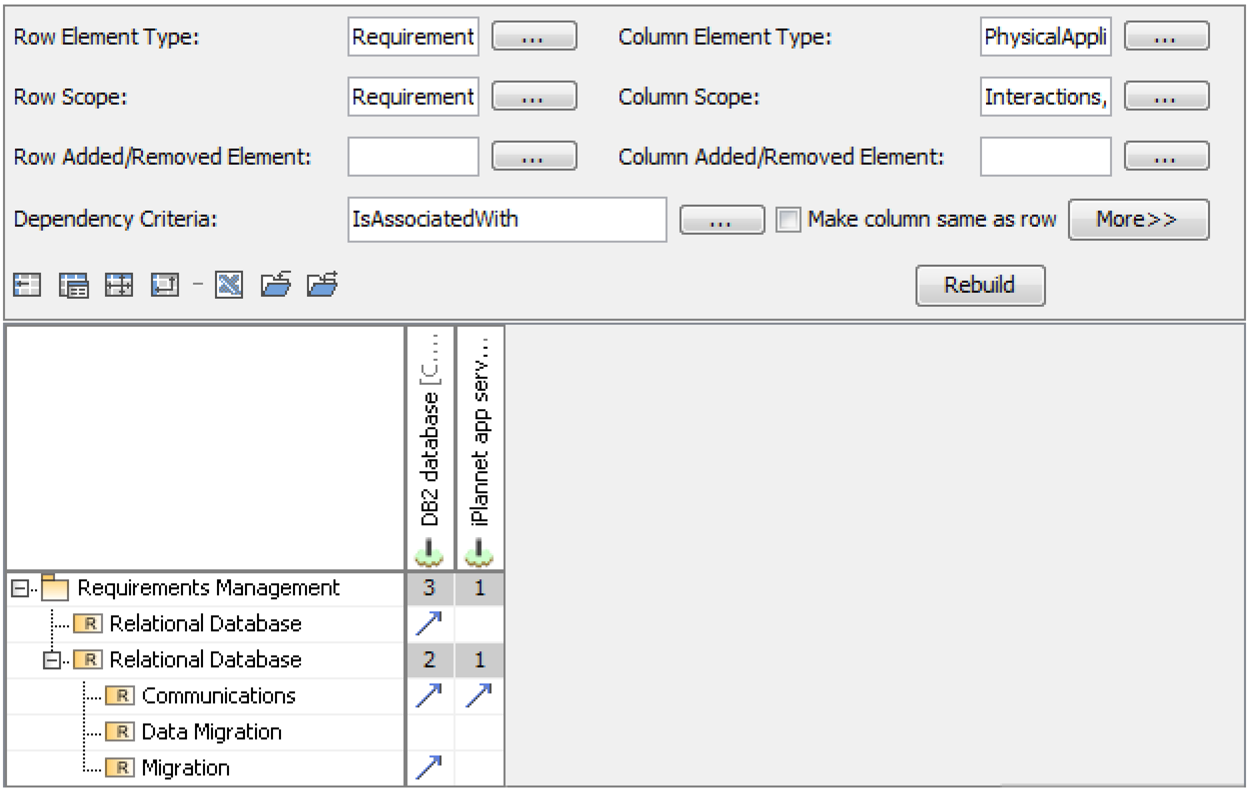
Let us say, we have requirements in the Requirements Management model, and we need to show what application components meet these requirements.

A dependency matrix will be used to display the Is Associated With relations between requirements and physical application components.

To display dependencies between the requirements and the application components:

1. Create a new dependency matrix.
2. Select Requirement as a row element type and Physical Application Component as a column element type.

3. Select Requirements Management as a row scope and the entire TOGAF model as a column scope.
4. Select Is Associated With as a dependency criteria.
5. Click **Rebuild**.



Row Element Type: Requirement
Column Element Type: PhysicalAppli
Row Scope: Requirement
Column Scope: Interactions,
Row Added/Removed Element:
Column Added/Removed Element:
Dependency Criteria: IsAssociatedWith
□ Make column same as row

	DB2 database [C....	iPlannet app serv...
Requirements Management	3	1
Relational Database	↗	
Relational Database	2	1
Communications	↗	↗
Data Migration		
Migration	↗	

Figure 9 -- Displaying dependencies between requirements and physical application components

For more information about using dependency matrixes, see “Dependency Matrix” in *MagicDraw UserManual.pdf*.

Using Generic Table

In order to review and edit property values of some selected TOGAF model elements in a single place, you can use generic table (tabular format), the feature introduced by MagicDraw. Using the tabular format is more convenient than for example browsing through Specification windows of the elements and editing their properties one by one.

Case study

Let us say, we need to edit the **Category** and **Statement Of Principle** property values for all Principle elements created in a TOGAF model.

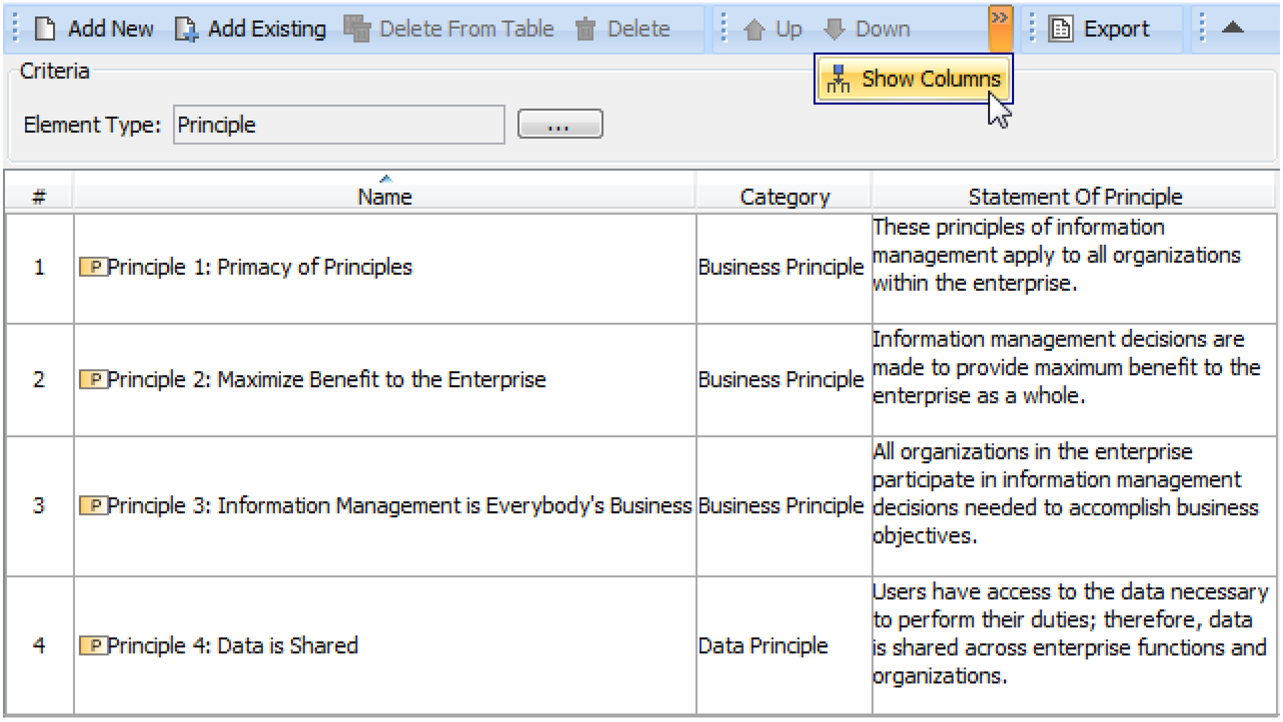
A generic table will be used to display all the Principle elements in your TOGAF model and their both **Category** and **Statement Of Principle** properties.

To create a generic table for principles:

1. Create a new generic table.
2. Under **Criteria**, click the **...** button next to the **Element Type** box.
3. In the **Select Element Types** dialog, select Principle and click **OK**.

4. On the generic table toolbar, click **Add Existing** and in the **Select Principle** dialog select all principle elements.
5. On the generic table toolbar, click **Show Columns** and select the **Category** and **Statement Of Principle** properties.

TIP! You can also use the **Generic Table Wizard** for creating a new generic table.



Criteria

Element Type: Principle

#	Name	Category	Statement Of Principle
1	P Principle 1: Primacy of Principles	Business Principle	These principles of information management apply to all organizations within the enterprise.
2	P Principle 2: Maximize Benefit to the Enterprise	Business Principle	Information management decisions are made to provide maximum benefit to the enterprise as a whole.
3	P Principle 3: Information Management is Everybody's Business	Business Principle	All organizations in the enterprise participate in information management decisions needed to accomplish business objectives.
4	P Principle 4: Data is Shared	Data Principle	Users have access to the data necessary to perform their duties; therefore, data is shared across enterprise functions and organizations.

Figure 10 -- Displaying selected properties of principles

For more information about using generic tables, see “Generic Table” in MagicDraw UserManual.pdf.

Extending TOGAF Model with UML

You can use UML elements to extend your TOGAF model.

Case study 1

Let us say, we need to detail a the “Claim damage” business process.

An activity diagram will be used to display the sequence of processes inside of the “Claim damage” process.

In order to create an activity diagram, right-click the “Claim damage” process and from the shortcut menu select **New Diagram** > **Activity Diagram**. In the created activity diagram you can use UML elements to model the sequence of processes inside the “Claim damage” process.

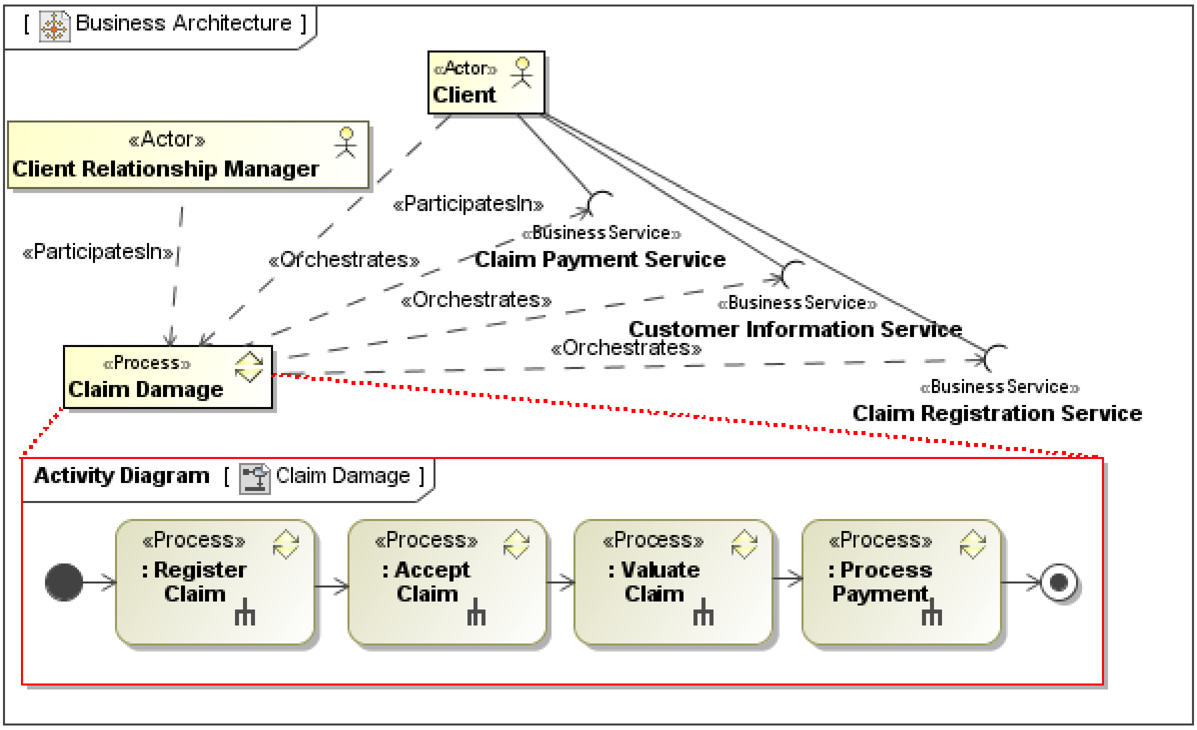

[Business Architecture]
«Actor» Client
«Actor» Client Relationship Manager
«ParticipatesIn»
«ParticipatesIn»
«Orchestrates»
«Orchestrates»
«Orchestrates»
«Business Service» Claim Payment Service
«Business Service» Customer Information Service
«Business Service» Claim Registration Service
«Process» Claim Damage
Activity Diagram [Claim Damage]
«Process» : Register Claim
«Process» : Accept Claim
«Process» : Valuate Claim
«Process» : Process Payment

Figure 11 -- “Claim damage” process detailed in UML activity diagram

Case study 2

Let us say, we need to detail the “Financial application” physical application component.

A composite structure diagram will be used to display the interaction of the components inside the “Financial application” physical application component.

In order to create a composite structure diagram, right-click the “Financial application” physical application component and from the shortcut menu select **New Diagram** > **Composite Structure Diagram**. In the created composite structure diagram you can use UML elements to model the interaction between components inside the “Financial application” physical application component.

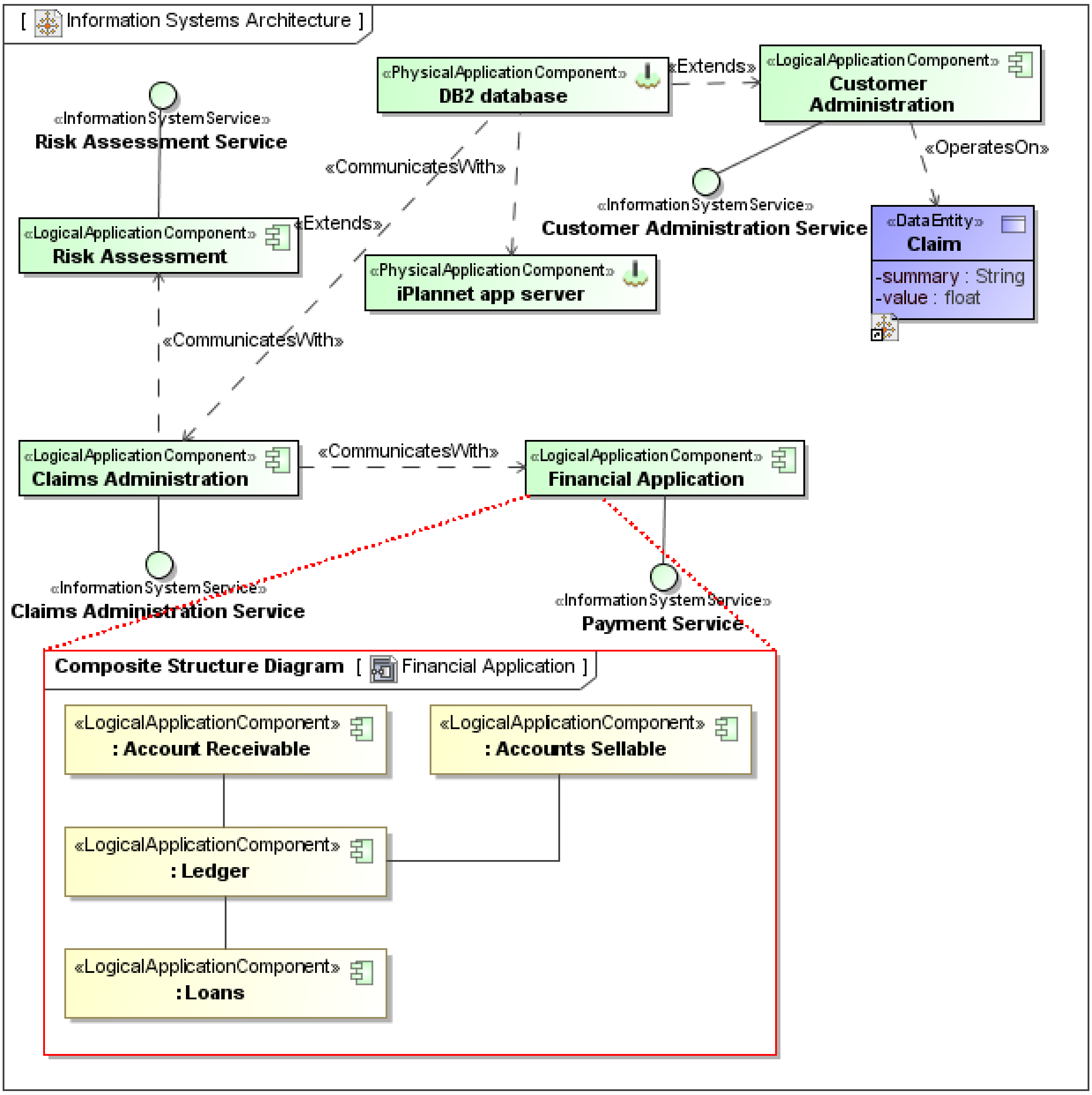

[Information Systems Architecture]
«InformationSystemService»
Risk Assessment Service
«PhysicalApplicationComponent»
DB2 database
«Extends»
«LogicalApplicationComponent»
Customer Administration
«CommunicatesWith»
«OperatesOn»
«InformationSystemService»
Customer Administration Service
«DataEntity»
Claim
-summary : String
-value : float
«LogicalApplicationComponent»
Risk Assessment
«Extends»
«PhysicalApplicationComponent»
iPlannet app server
«CommunicatesWith»
«LogicalApplicationComponent»
Claims Administration
«CommunicatesWith»
«LogicalApplicationComponent»
Financial Application
«InformationSystemService»
Claims Administration Service
«InformationSystemService»
Payment Service
Composite Structure Diagram [Financial Application]
«LogicalApplicationComponent»
: Account Receivable
«LogicalApplicationComponent»
: Accounts Sellable
«LogicalApplicationComponent»
: Ledger
«LogicalApplicationComponent»
: Loans

Figure 12 -- “Financial application” logical application component detailed in UML composite structure diagram