



Enterprise Architecture Management Tool Survey 2014 Update

Florian Matthes, Matheus Hauder, Nikolaus Katinszky

Software Engineering for Business Information Systems (sebis)
Chair for Informatics 19
Technische Universität München
Boltzmannstraße 3, 85748 Garching b. München, Germany

About

Software Engineering for Business Information Systems (sebis) is a research chair at the Institute for Informatics of the Technische Universität München. *sebis* has been established in 2002 with funding of the Ernst Denert-Stiftung and is headed by Professor Dr. Florian Matthes. Main research of *sebis* are:

- *Enterprise Architecture Management*: Development of methods that support the strategic planning, analysis, and enactment of holistic models for business processes, information systems, and IT infrastructure with their relationships.
- *Social Software Engineering*: Development of social software solutions that improve the collaborative organization of information and processes.

Copyright

Copyright © 2014 Technische Universität München, *sebis*, Germany. All rights reserved. No part of this publication may be reproduced, stored, archived, or transmitted in any form by any means (electrical, mechanical, by photocopying, or otherwise) without the prior printed permission of the publisher, Technische Universität München, *sebis*. The information contained and opinions expressed herein may be changed without prior notice.

Trademarks

All trademarks or registered trademarks are the property of their respective owners.

Disclaimer

The information contained herein has been obtained from sources believed to be reliable at the time of publication but cannot be guaranteed. Please note that the findings, conclusions and recommendations that TU München, *sebis* delivers will be based on information gathered in good faith from both primary and secondary sources, whose accuracy cannot always be guaranteed by TU München, *sebis*.

TU München, *sebis* disclaims all warranties as to the accuracy, completeness, or adequacy of such information. TU München, *sebis* shall have no liability for errors, omissions, or inadequacies in the information contained herein or for interpretations thereof. In no event shall TU München, *sebis* be liable for any damage of any kind (e.g. direct, indirect, special, incidental, consequential, or punitive damages) whatsoever, including without limitation lost revenues, or lost profits, which may result from the use of these materials. The reader assumes sole responsibility for the selection of these materials to achieve its intended results.

There will be no warranty that the results of this publication will be economically and technically exploitable and unencumbered by third-party proprietary rights.

Abstract

Over the past decade Enterprise Architecture (EA) developed to a mature discipline that is widely applied in many organizations. EA departments are firmly established in large organizations helping various stakeholders in the business and the IT departments to get a better understanding of the complex interrelationships of business processes, information systems and IT infrastructure. Main goal of these initiatives is to manage to evolution of the EA. For this purpose enterprise architects document current and future states of the EA in repositories.

Due to the complexity of organizations today many tool vendors provide software solutions for the management of these repositories. Tools for EA management support enterprise architects with the documentation, the generation of reports as well as the communication with stakeholders. The variety and the number of tools that are available for EA management on the market makes the selection of the right tool increasingly difficult for organizations. Next to well established tool vendors many smaller products are entering the market with niche products recently.

In this study we investigate EA management tools with a scenario based approach. These scenarios are evaluated with a consistent information model of a fictitious organization. Every step of these scenarios is well described and documented with screenshots to help readers comprehend the capabilities of the tools. This information model as well as the scenarios are based on the our previous EA management tool survey that was published six years ago in 2008. Although minor criterias for the assessment have changed over the time, we experienced that the general scenarios are still valid today.

Based on our previous EA management tool survey, we created an update with four new tools that were not investigated previously. The evaluation of each tool is categorized into two parts: One part deals with specific functionality, e.g. *adapting the information model, supporting multiple users and collaborative work, creating visualizations of the application landscape, or usability*. The second part analyzes the enterprise architecture management support of the tools, e.g. *landscape management, project portfolio management, application architecture management, SOA transformation*.

Contents

Abstract	iii
1 BiZZdesign Architect	2
1.1 Evaluation of Specific Functionality	2
1.2 Evaluation of EA Management Support	12
2 LeanIX	21
2.1 Evaluation of Specific Functionality	21
2.2 Evaluation of EA Management Support	30
3 SAP Sybase PowerDesigner	37
3.1 Evaluation of Specific Functionality	38
3.2 Evaluation of EA Management Support	50
4 Iteraplan	57
4.1 Evaluation of Specific Functionality	57
4.2 Evaluation of EA Management Support	68

In this chapter the previous mentioned EA scenarios will be applied on the software BiZZdesign Architect.

Architect is developed by the dutch company BiZZdesign. It is available as a fat client version that can be installed on any computer using Windows XP or higher. An object repository can also be stored on a central server that is accessed by clients. These clients still need to have the full version installed locally on their device.

We use the Architect 4.3.2 trial version that offers the same features as the full version in our evaluation. After registering at BiZZdesign's website¹, the demo can be downloaded and used for 30 days free of charge.

1.1 Evaluation of Specific Functionality

In this section specific EAM functionality are evaluated.

1.1.1 Importing, Editing and Validating

Architect offers the option to import whole models through XML files or Microsoft Visio diagrams e.g. for ArchiMate. The Visio files need to be present as Visio XML files (.vdx). Then, Visio diagrams can be integrated into Architect models and new objects can be imported into the Architect repository. For the XML import the source XML file needs to be transformed to standard Architect XML exchange format by using an manually written XSLT transformation file (Fig. 1.1). Afterwards the resulting new XML file can be imported.

¹<http://www.BiZZdesign.com/downloadmanager/download/19>

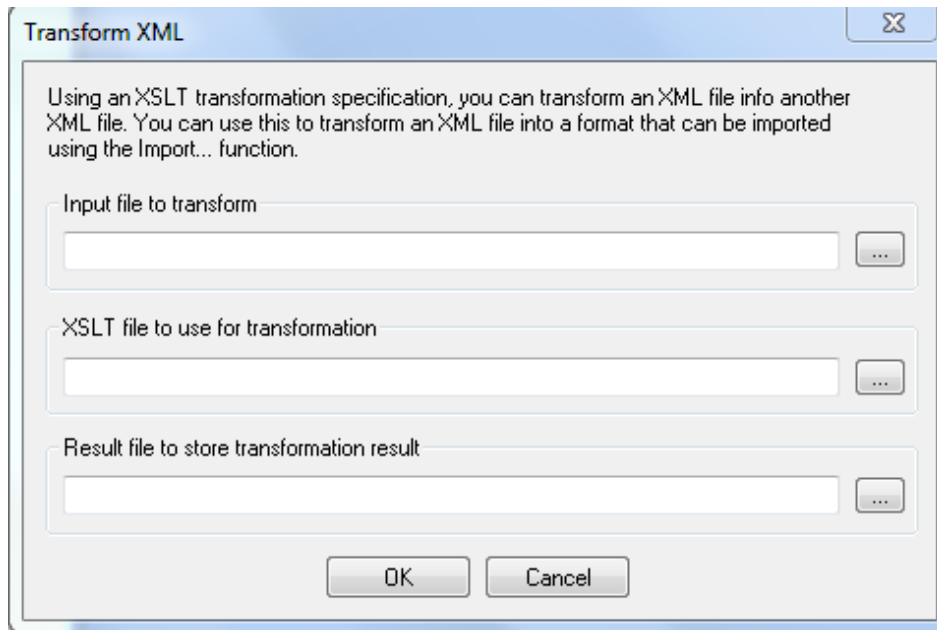


Figure 1.1: Architect - XML Import Window

Another quick way to import import objects is to copy data from Excel or Word sheets into cache and paste them into a table while creating new multiple objects in Architect (Fig. 1.2).

Architect models are saved as proprietary .xma files that follow a XML structure. They can be exported as XML files, SQL scripts that create tables and .csv files.

After the successful import object relations can be set easily using the Cross Relations Table. In a pop-up window the relation and the objects to be connected can be selected first (Fig. 1.3). Then, the selected relations can be set by simply doing ticks in a matrix (Fig. 1.4).

The set of predefined properties vary depending on the object type. For example, work package objects that can be treated as projects have predefined attributes for costs, start and end date, while application objects only have 'Related objects' as predefined attribute. Moreover, there are property profiles that can be regarded as sets of attributes. They can be activated to increase the amount of available attributes. In this way, statuses about the life cycle like 'In development' or 'In use' can be set. It is also possible to add profiles with manually created attributes. Thus, own attributes can be added. In order to add own profiles, one need to manually create 3 files (2 .mps files and 1 .mpd file). that include attribute definitions in an XML-like language. The last described method of creating .mps and .mpd files for creating new attributes has not been evaluated.

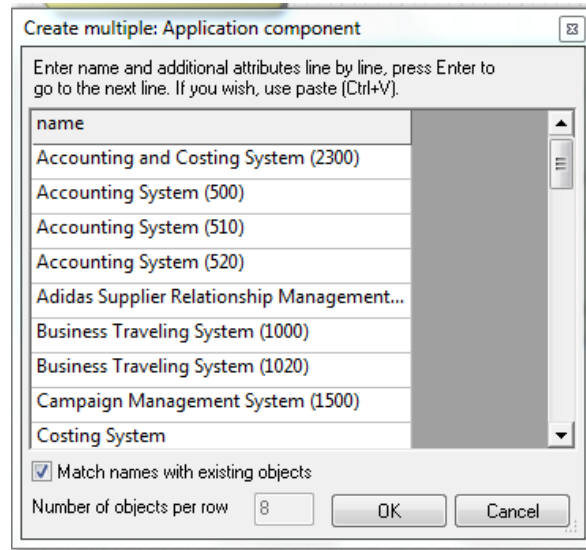


Figure 1.2: Architect - XML Import Copy Paste

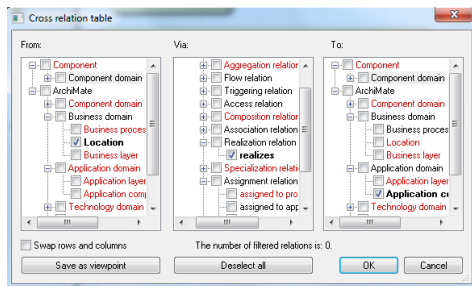


Figure 1.3: Architect - Cross Relation Table 1

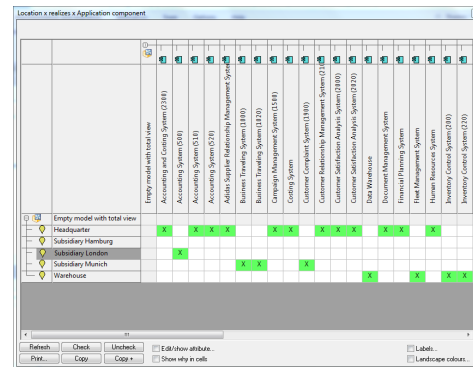


Figure 1.4: Architect - Cross Relation Table 2

1.1.2 Creating visualizations

Architect offers two ways to create visualizations: drawing new diagrams manually in the model canvas or automatic generation using existing objects from the repository. In order to ease the creation process the automatic generation feature was mainly evaluated.

Architect makes use of a view concept. There are different views that allow different objects to be used. Views are aligned with business layer, application layer and technology layer from the ArchiMate domain. Examples of different views are: total view, application behavior view, implementation and deployment view, or layered view.

For generating a view one first has to select the objects to be displayed (e.g. business processes). Then the type of view can be set, which relations shall be represented and which object types shall be integrated into the view (Fig. 1.5). A new visualization is generated that can be edited afterwards. Many different visualizations can be generated, e.g. cluster maps (Fig. 1.6), graph layout maps (Fig. 1.7) or process support maps (Fig. 1.8).

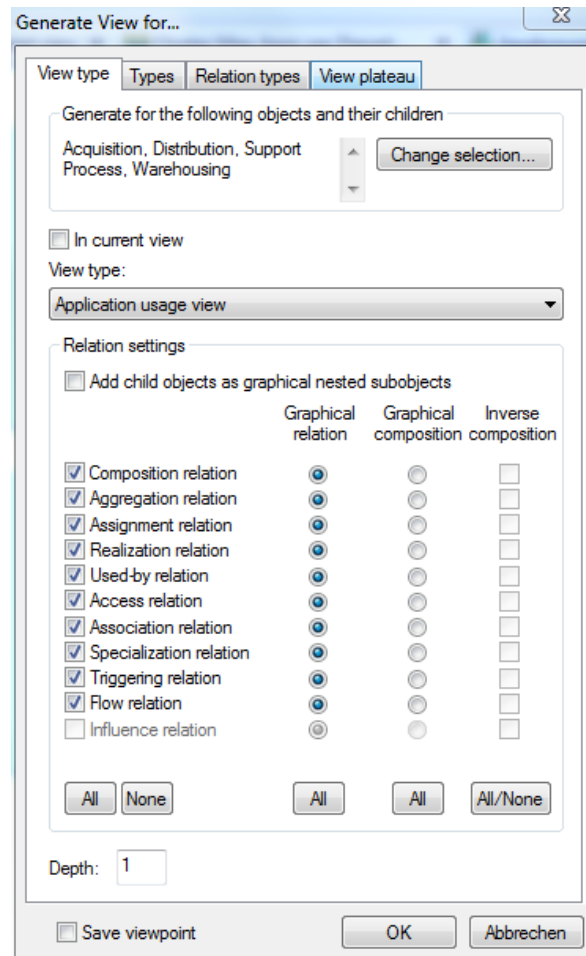


Figure 1.5: Architect - View Generation

Diagrams involving a time dimension like time interval maps are not available. Development over time can be regarded through plateaus (time intervals) that reflect changes. A portfolio diagram is not available. Only relations between objects can be visualized. Attributes like costs or urgency can only be reflected through view filters that will be explained in the next part.

1.1.3 Interacting with, Editing of, and Annotating Visualizations

After visualizations have been generated, Architect offers many possibilities to edit and interact with them. Size and position of objects can be changed as well as font style, icons and transparency level. Color highlighting and labels for annotations exist, too. There are auto-layout and zoom in / zoom out features available. It is possible to jump into nested diagrams to view further details.

There are view filters that allow to highlight specific relations through color coding, labels or tooltips (Fig. 1.11 and Fig. 1.12). These view filters can be saved as viewpoints that can be

1. BiZZdesign Architect

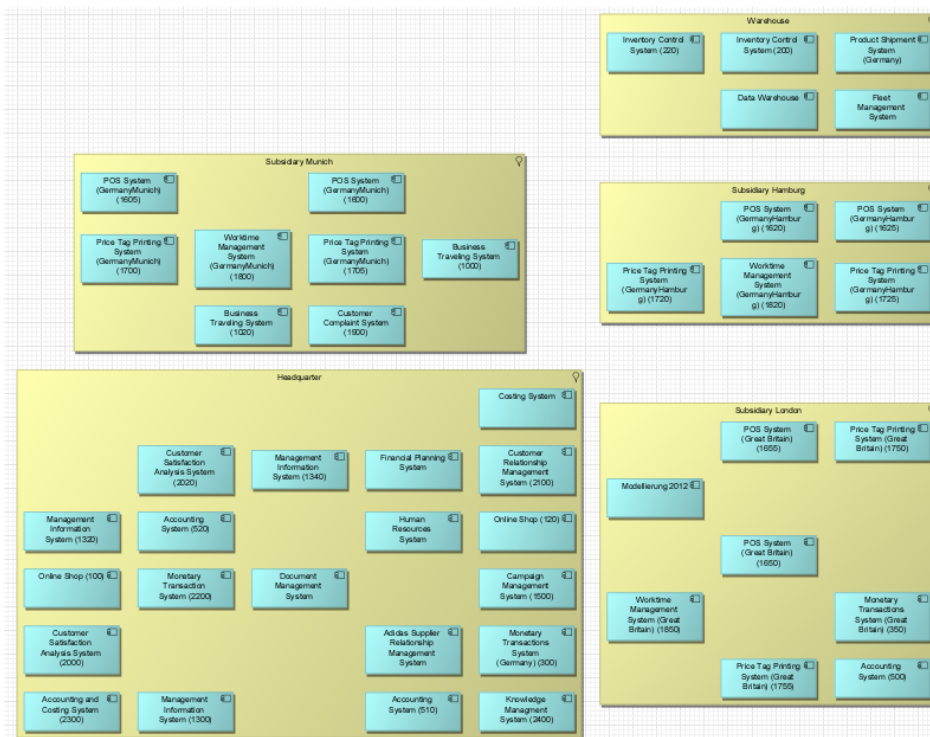


Figure 1.6: Architect - Cluster Map

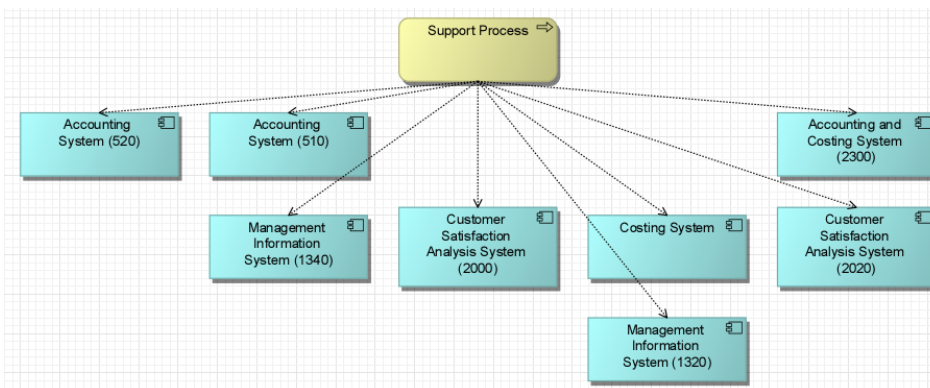


Figure 1.7: Architect - Business Process uses Applications

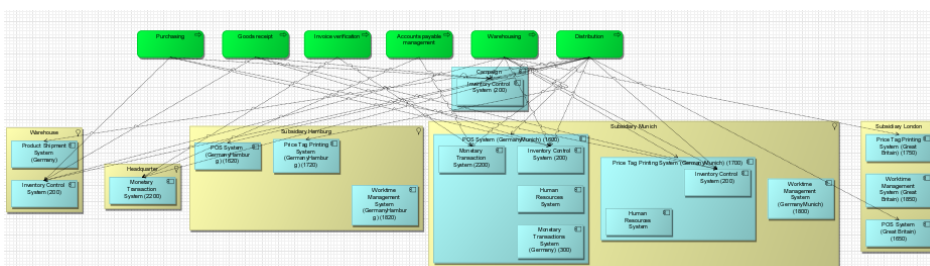


Figure 1.8: Architect - Process support Map

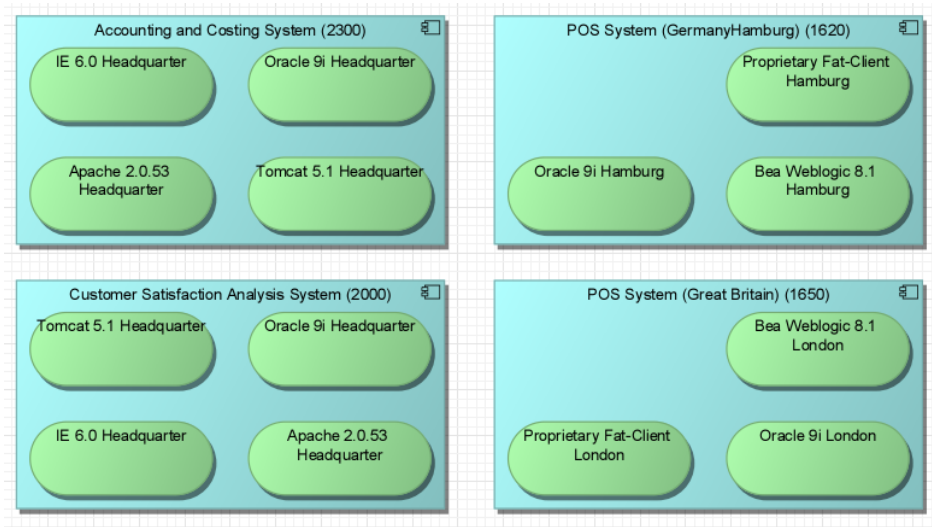


Figure 1.9: Architect - Applications use Infrastructure Services

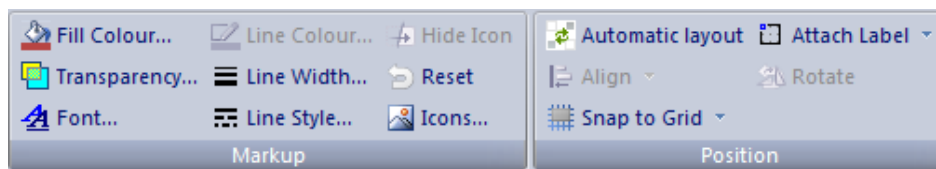


Figure 1.10: Architect - Visualization interaction options

activated and deactivated manually for each visualization. Viewpoints can be saved for later use. These view filters are useful to combine different views or information.

1.1.4 Communication and Collaboration Support

In Architect models can be saved and imported by other users. There are also shared model packages (.sma files) that can be opened by multiple users at the same time. Visualization changes are stored to user specific work spaces and cannot be seen by other users. Model packages can be separated into versions and offer SVN-like features to check in and check out models, views or objects (Fig. 1.13). When checking in a model, differences to the central model are displayed.

There is also the possibility to store models in a database-like repository on a central server. It shall be possible to assign different roles for read, write and manage access rights. The shared view shall represent differences between revisions by different users: objects differing from the shared model are highlighted. This feature has not been evaluated. Modern collaboration features like chats are not available.

1. BiZZdesign Architect

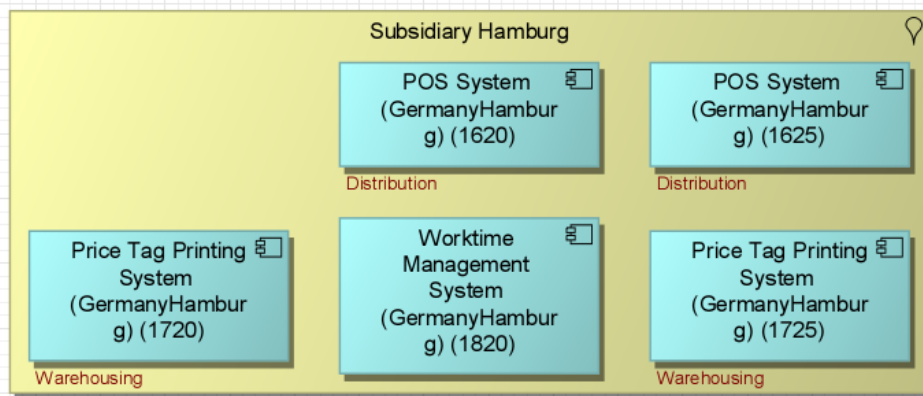


Figure 1.11: Architect - Label view applied

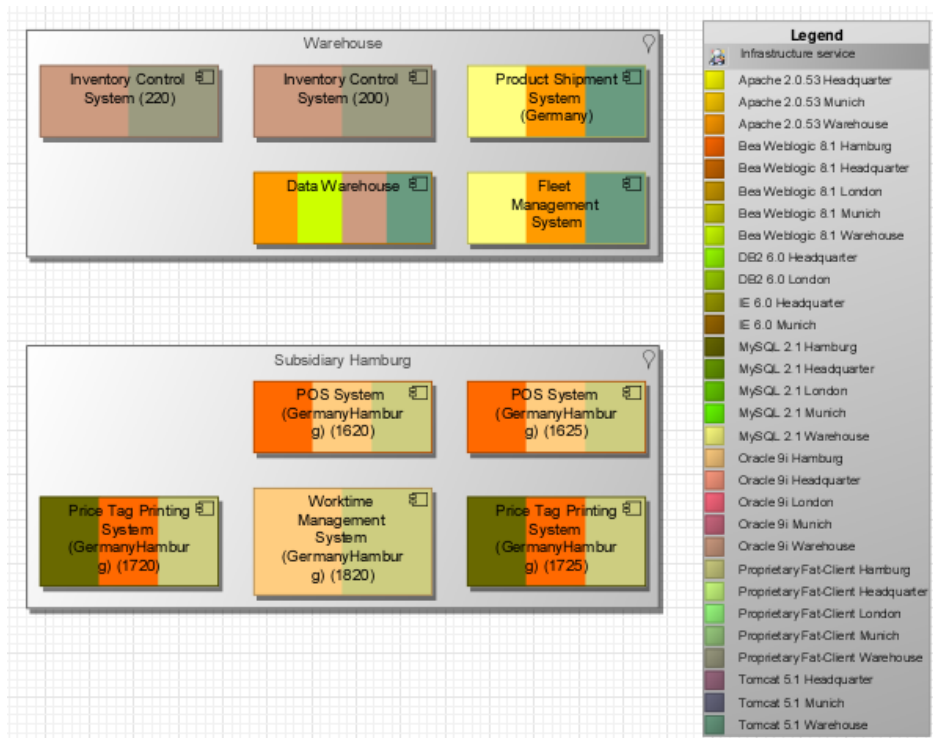


Figure 1.12: Architect - Color view applied

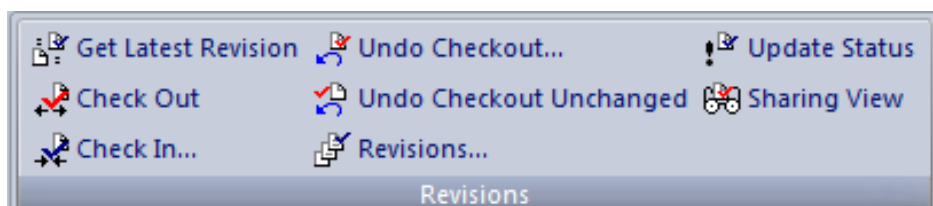


Figure 1.13: Architect - Collaboration Functionality

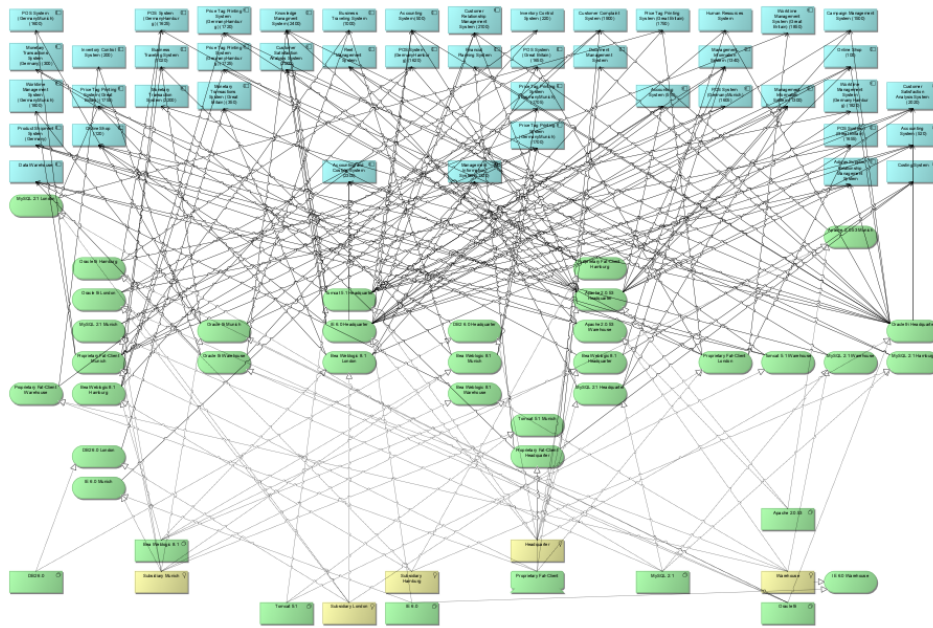


Figure 1.14: Architect - Overview visualization

1.1.5 Flexibility of the Information Model

Architect is aligned to the ArchiMate information model that is based on business, application and technology layer. This information model includes structural and behavioral relations and a huge list of object types. Moreover, there is the motivation extension to support strategy planning and modeling. Architect is Archimate 2.1 certified. There the Archimate model is quite strict implemented. New object types, relations or attributes cannot be added easily. As mentioned earlier, additional object attributes can be added by creating or editing new profiles through .mpd files. According the Architect manual different attribute types like strings, integers or dates and predefined values can be set. The creation of new attributes has not been evaluated.

1.1.6 Support of large scale Data

While evaluating Architect it could be noticed that browsing through the repository to look for particular items can be difficult within a huge amount of objects. With an increased amount of applications and other data this process would even more difficult. To avoid such problems the tool provides a search function on the repository browser in case

While auto-generating a bigger visualization (like Fig. 1.14) with around 50 applications and less than 400 relations delays could be noticed. Architect did not crash but did not react for some seconds until the visualization was generated. It might be that these delays would be reduced if Architect was running on a more powerful computer, though. Because of these delays the creation of additional test applications and relations has been skipped. While auto-generating smaller visualization Architect did not show any delays.

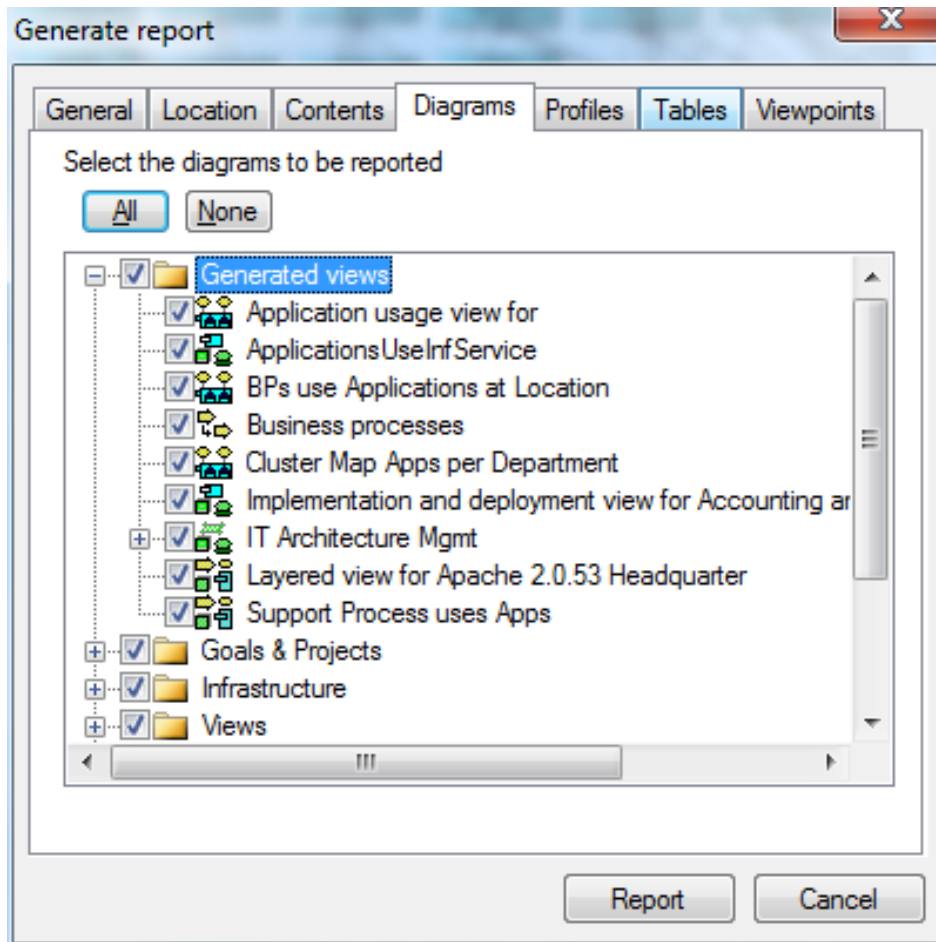


Figure 1.15: Architect - Report generation

1.1.7 Reporting

Architect allows to automatically generate reports in HTML and RTF format. These reports allow to illustrate the visualizations that were created in Architect to a broader audience. The design template the HTML report is based on, the content and the diagrams to be integrated can be changed and adjusted (Fig. 1.15). An automatically generated HTML report using the standard template can be seen in Fig. 1.16. Inside the HTML report there is report manual that is helpful because the navigation and icons are not always intuitive.

Visualizations generated in Architect can be exported as .pdf, .svg and .emf files as well. Multiple objects, e.g. a list of applications, can be exported into Excel sheets using the Excel export feature. Word files can be automatically generated that include all created visualizations in one Word document. Architect offers the possibility to automatically generate a Powerpoint presentation where each diagram is displayed as an image on one slide each (Fig. 1.17). This feature is useful to integrate quickly visualizations from Architect into presentations.

Moreover, Architect offers a powerful query language that can be used to extract data and

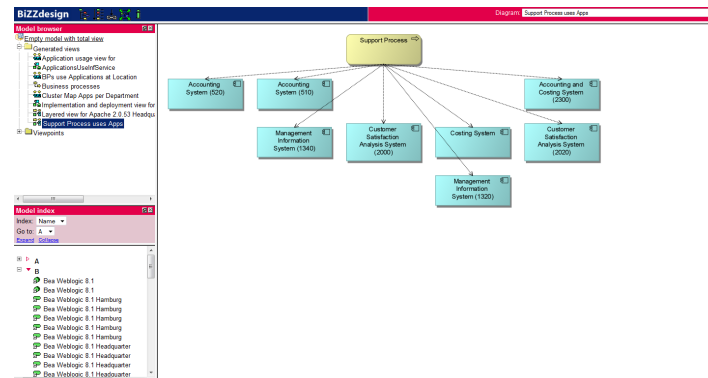


Figure 1.16: Architect - HTML report

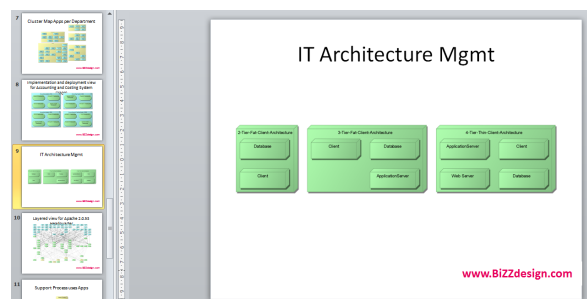


Figure 1.17: Architect - Powerpoint report

define how output is presented. According to the manual logical operations, data types, variables, iterations, conditions and methods can be used. The query language has not been evaluated in detail.

1.1.8 Usability

Architect allows to generate and edit visualizations easily and in a smooth way. The tabbed menu in the top area of the application reminds of Microsoft office products and helps the user to learn about the application quickly.

The view generation windows is helpful to make a pre-selection of the objects and relations to be displayed. Unfortunately, there is no visualization preview available. The visualization is generated immediately after setting the options. If the freshly generated visualization is not the intended by the user, it can be edited, or a new needs to be generated. Views that are not important or not useful anymore need to be deleted manually. Otherwise one will end up in too many views and loose the overview. Sometimes it is difficult to find views that were generated earlier because the automatically view names are not always meaningful. As mentioned earlier, the view generation process can become slow when a visualization with many objects or relations is being generated.

The overview is sometimes hard to keep because of many small windows inside the application. The search feature in the object browser is useful. Without this feature, it is difficult to find

1. BiZZdesign Architect

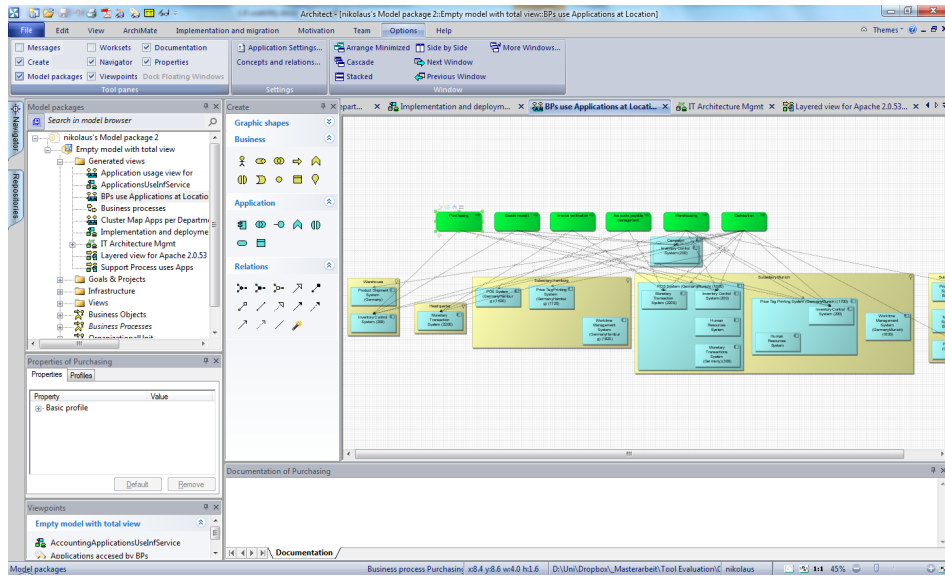


Figure 1.18: Architect - Application overview

specific objects. Viewpoints are helpful to blend in specific information into visualizations. The integrated legend helps as well.

The manual that comes with the installation is partly helpful. It is a general guideline and does not answer detailed questions perfectly .

1.2 Evaluation of EA Management Support

In this section BiZZdesign Architect's EA Management support capabilities are evaluated.

1.2.1 Landscape Management

Application visualizations can be used to support landscape management. Visualizations like cluster maps (Fig. 1.6) and layered views (Fig. 1.18) show connections to business and technology layer. Architect offers predefined attributes to set a life cycle status: 'in development', 'in use', 'out of date' and 'out of use'.

Change over time and landscape planning can be performed using Architect's roadmap functionality. This concept allows to plan landscape development for different points in the future, as well as multiple planning scenarios for one point in the future. One roadmap consists of several plateaus where each plateau represents a time interval. Fig. 1.19 and Fig. 1.20 illustrate differences in the current and planned landscape. The concept of roadmaps and plateaus allows flexible landscape planning taking into account different planning scenarios. The evolution of the roadmap can be highlighted, e.g. by activating color filters.

	Current Landsc	Planned Landsc
Accounting System (500)	V	
Accounting System (510)		V
Adidas Supplier Relationship ...	V	V
Business Traveling System (10...	V	
Business Traveling System (10...		V
Campaign Management Syste...	V	V
Costing System	V	V
Customer Complaint System (...)	V	V
Customer Relationship Mana...	V	V

Figure 1.19: Architect - List of current and planned applications

1.2.2 Demand Management

Demands can be modeled in Architect using Archimate's motivation extension. The object type 'Requirement' can reflect IT- and business-related demands. These demands can be linked to strategic goals, projects and applications. Thus, ongoing projects can realize demands that impact long term goals.

1.2.3 Project Portfolio Management

Projects are integrated into Architect with so-called 'Work packages'. Multiple work packages can be composed to one program. They can be connected to business processes, applications, goals and demands (Fig. 1.22). Start/end dates, costs and related actors are predefined attributes.

A project portfolio matrix using attributes as dimensions is not available but projects can be modeled in views and linked to various objects. One option to integrate attributes like costs into visualizations is to use labels, tooltips or color coding (Fig. 1.23).

Sub-projects and relations between projects can be modeled in Architect. Time conflicts of overlapping are not recognized in Architect.

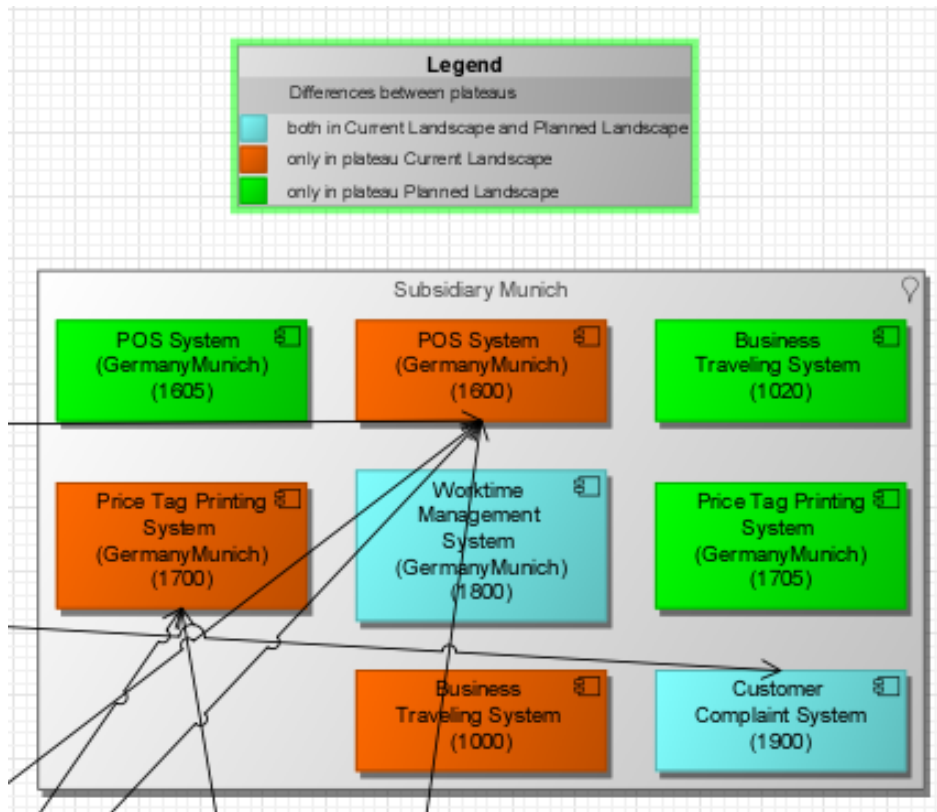


Figure 1.20: Architect - Visualization of current and planned applications

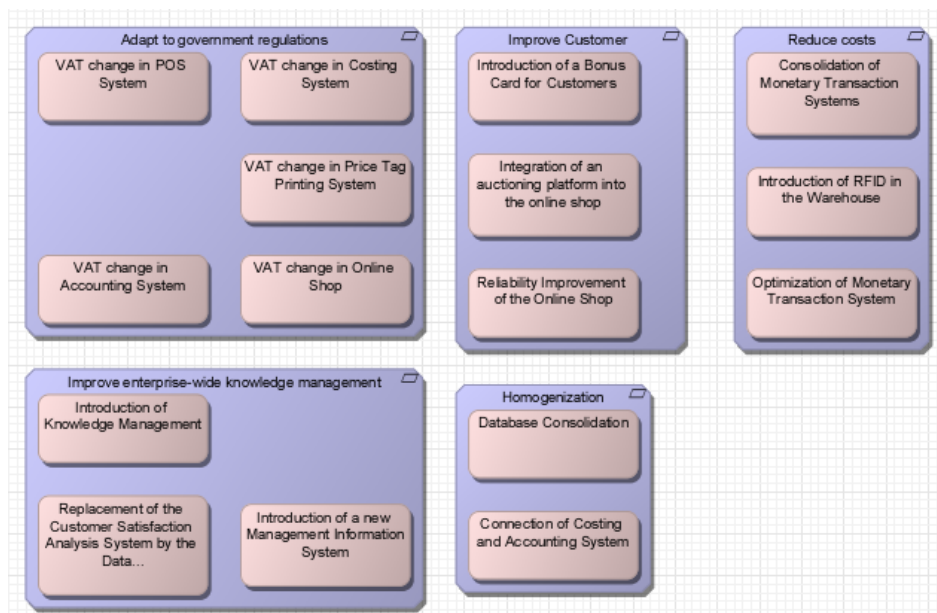


Figure 1.21: Architect - Projects linked to demands

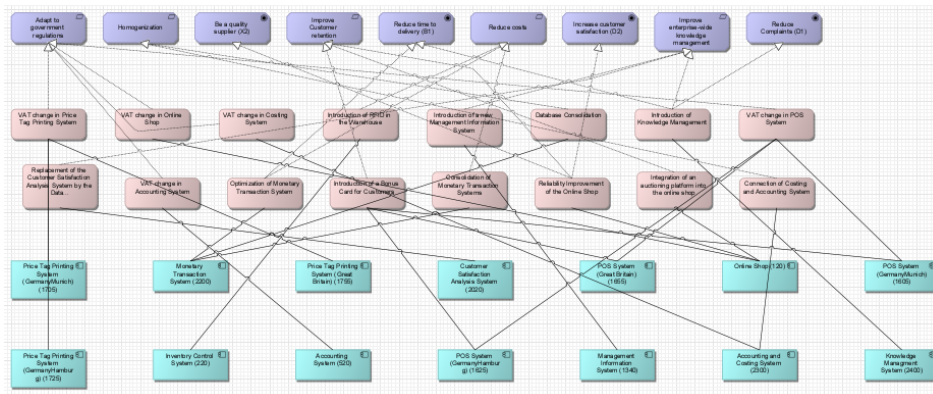


Figure 1.22: Architect - Projects linked to demands and applications

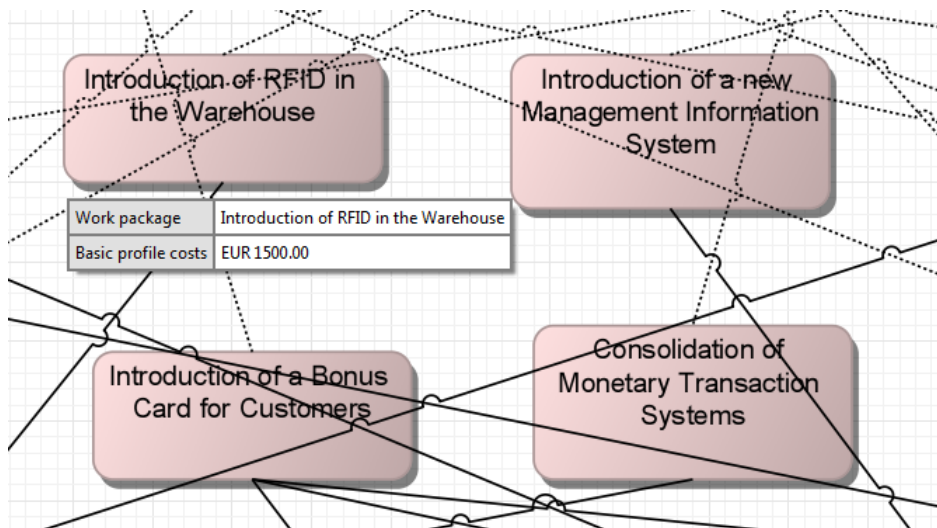


Figure 1.23: Architect - Project costs as tooltip

1.2.4 Synchronization Management

Architect offers the possibility to model and visualize dependencies between projects. Organizational units or business processes that are affected by projects can be illustrated using layered views (Fig. 1.24).

Attributes like end dates or costs are not linked to dependent projects in Architect. Project delays can be incorporated by changing the end date of a project, but these delays do not affect dependent projects. A hint or alarm that reminds of the dependency is not available. Visualizations that integrate the dimension time like Gantt charts do not exist in Architect. The only way to integrate time intervals into visualizations is to use plateaus.

1. BiZZdesign Architect

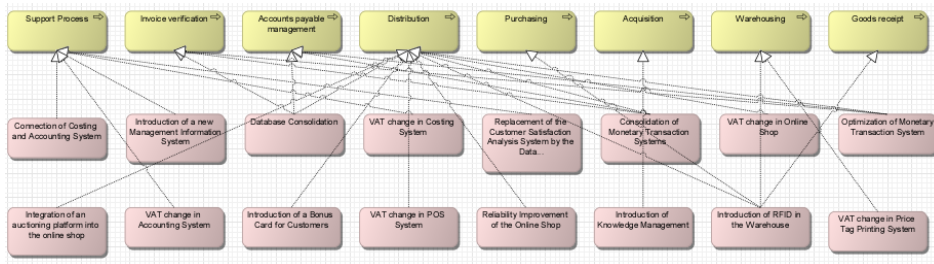


Figure 1.24: Architect - Business processes affected by projects

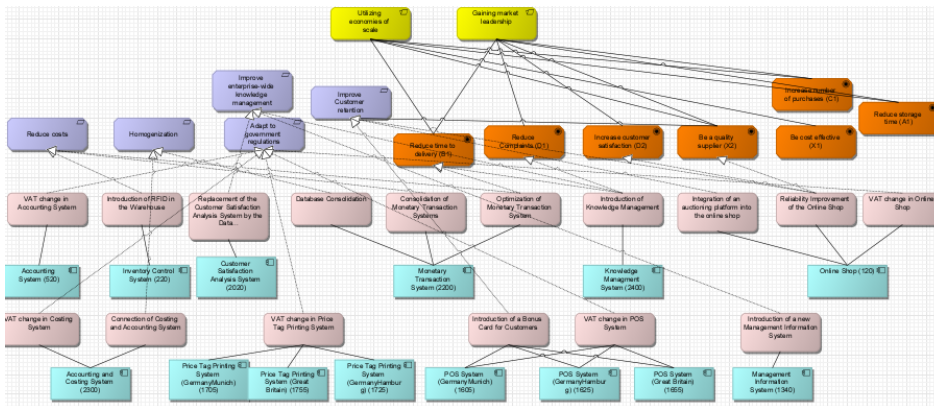


Figure 1.25: Architect - Drill down from strategies to affected applications

1.2.5 Strategies and Goals Management

Goals are available through the motivation extension in Architect. A strategy object type is not available but strategies can be integrated into Architect using the object type "Program". Multiple projects can be comprised to one Program. A layered view can be used to drill from strategies over goals and projects down to applications (Fig. 1.25). This view is useful to perform an impact analysis and see which applications are affected by which projects or strategies. Paths on a way down can also be highlighted.

Tabular reports that group projects to goals or strategies are not available. Moreover, a goal fulfillment rate can not be represented in Architect.

1.2.6 Business Object Management

Business objects are integrated into Architect through explicit object types. It can be modeled which business processes, applications or infrastructure objects use particular business objects. It is not possible to specialize which operation (e.g. CRUD) can be performed on business objects. The only possibility to integrate them is to write in the documentation field of the corresponding relation or use labels in visualizations.

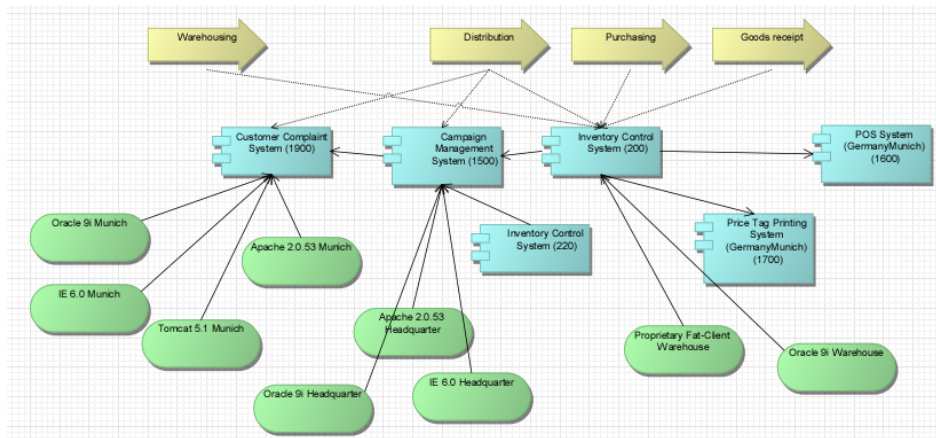


Figure 1.26: Architect - Service oriented architecture diagram

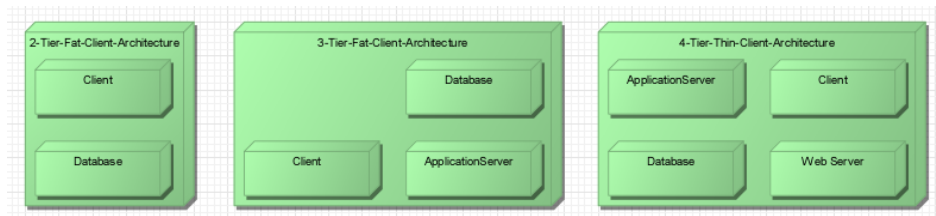


Figure 1.27: Architect - Architectural blueprints

1.2.7 SOA Transformation

The concept of business, application and infrastructure services is integrated into Architect. Potential applications to be replaced by services can be identified to architecture overview views. There is also a 'Service realization view' existent in Architect. Unfortunately, this view allows only to model business and application services, not infrastructure. Therefore, the total view is more useful to see the full picture of business, application and infrastructure layers (Fig. 1.26).

Tabular reports and reports that show the SOA development over time are not available. Architecture's roadmap concept can be used for SOA transformation planning and highlight changes.

1.2.8 IT Architecture Management

Architecture management can be performed in Architect using the wide amount of infrastructure object types. Blueprints can be planned and modeled in Architect using 'Node' object types that represent blueprint elements and blueprints (Fig. 1.27).

Modeling infrastructures is quite good in Architect. The use of planning variants is also possible through roadmaps. Visualizations show which applications use which types of infrastructure and whether they stick to one blueprint (Fig. 1.28).

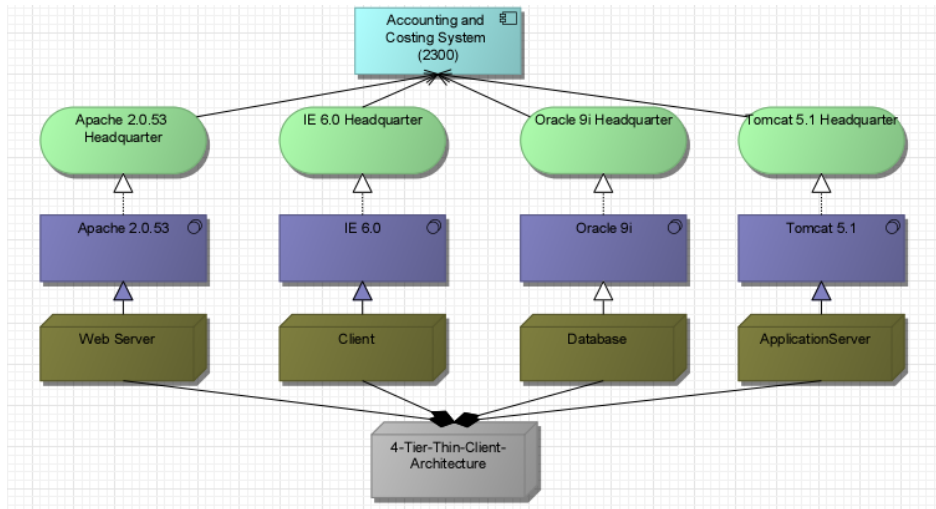


Figure 1.28: Architect - Accounting application uses infrastructure

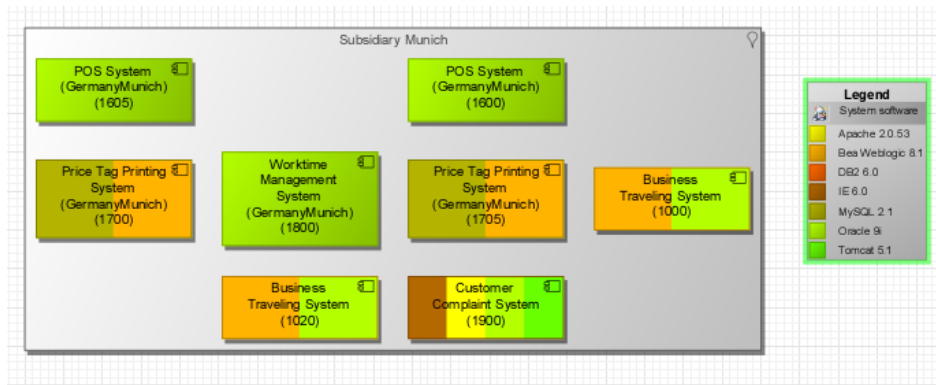


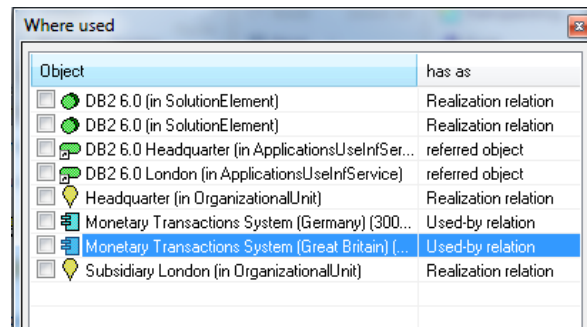
Figure 1.29: Architect - Application diagram with infrastructure view filter

The property tables as well as the cross reference tables in the tool provide simple tabular reports.

1.2.9 Infrastructure Management

There are infrastructure elements in Architect that allow relations to infrastructure services, blueprint elements and architectural blueprints. Multiple views can be used for infrastructure modeling: implementation and deployment view, infrastructure usage view an infrastructure view. Therefore, different layers and connections with infrastructure can be illustrated. View filters can be used to integrate infrastructure information into existing visualizations (Fig. 1.29).

Unfortunately, only visualizations and no tabular reports are present in Architect. There is a feature to show a list that includes information about where one specific object is used (Fig.



The screenshot shows a window titled "Where used" with a table of objects and their relationships. The table has two columns: "Object" and "has as". The objects listed are:

Object	has as
<input type="checkbox"/> DB2 6.0 (in SolutionElement)	Realization relation
<input type="checkbox"/> DB2 6.0 (in SolutionElement)	Realization relation
<input type="checkbox"/> DB2 6.0 Headquarter (in ApplicationsUseInfSer...)	referred object
<input type="checkbox"/> DB2 6.0 London (in ApplicationsUseInfService)	referred object
<input type="checkbox"/> Headquarter (in OrganizationalUnit)	Realization relation
<input type="checkbox"/> Monetary Transactions System (Germany) (300...)	Used-by relation
<input checked="" type="checkbox"/> Monetary Transactions System (Great Britain) [...]	Used-by relation
<input type="checkbox"/> Subsidiary London (in OrganizationalUnit)	Realization relation

Figure 1.30: Architect - Database utilization

1.30). Development and changes over time can only be modeled using Architect's roadmap concept. This concept to show infrastructure changes over time in visualizations.

In this chapter the previous mentioned EA scenarios will be applied on the software leanIX.

LeanIX is a rather new EAM tool developed by the German company LeanIX GmbH. The vendor emphasizes on its website the collaboration features, a simple user interface and reduced complexity.

In our evaluation we used the online trial demo version 1.10.0 that can be tested free of charge for 4 weeks.¹ It is provided as a web-based service and can be accessed through the browser.

2.1 Evaluation of Specific Functionality

In this section specific EAM functionality are evaluated.

2.1.1 Importing, Editing and Validating

LeanIX offers the possibility to import objects through predefined Microsoft Excel templates. After data have been inserted into a spread sheet, they can be validated by performing a test import, or be imported directly into the system. When performing the test run, a detailed spreadsheet is generated that shows which objects could be successfully imported or which errors occurred (Fig. 2.1).

Data can be exported from the system into Microsoft Excel spreadsheets. In these generated spreadsheets mass updates can be performed for a limited amount of attributes like 'Name', 'Description', 'Lifecycle' or 'Release'. After performing a mass update in Microsoft Excel, data can be again imported into leanIX. Interfaces and other important object attributes cannot

¹You can register for the online trial here: <https://leanix.net/en/site/contact?type=trial>.

2. LeanIX

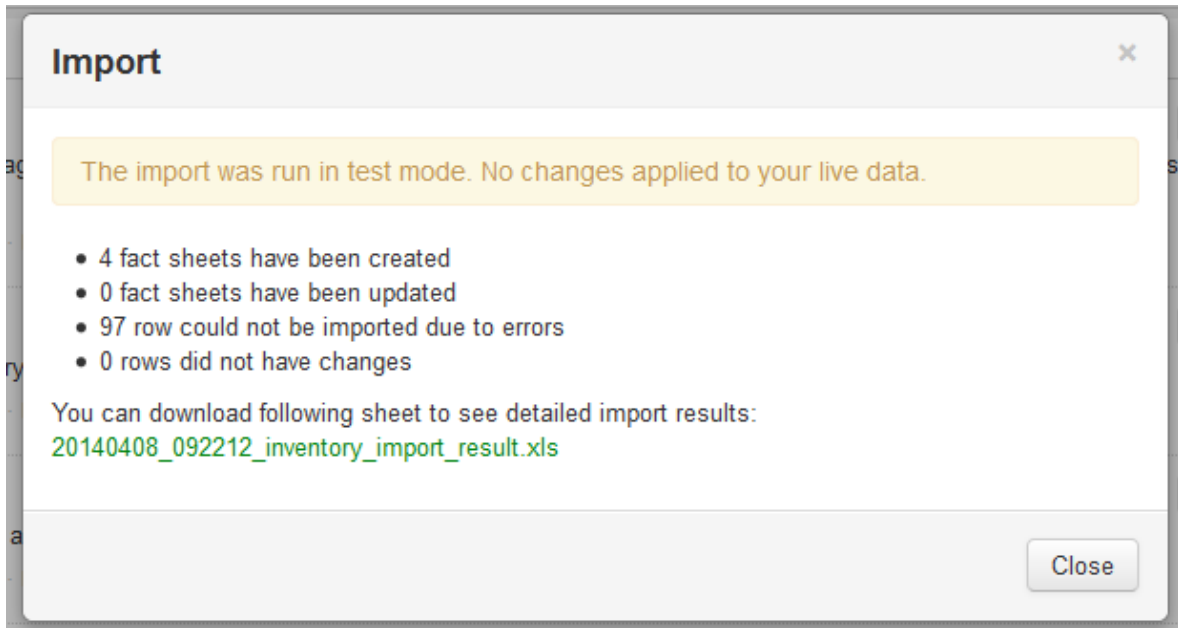


Figure 2.1: LeanIX - Import test-run

be set via mass updates but need to be changed in every object manually. This feature can be performed through a REST interface or additional scripts that are provided by the tool vendor for power users.

Objects can be edited in leanIX in the 'Inventory' tab. Each attribute can be edited either by enter individual text or using drop-down menus that shows potential input options (Fig. 2.2).

USED APPLICATIONS Which Applications are used by this Org Unit?

Application	# Users	Org Unit Fit
Accounting and Costing System (2300) x	<input type="text"/>	Please select
	Please indicate roughly how many users are using this Application.	★ ★ ★ ★ Unreasonable: Not enough or wrong functionality.
		★ ★ ★ ★ Insufficient: Rudimentary functional support.
		★ ★ ★ ★ Appropriate: All major functions.
		★ ★ ★ ★ Perfect: High number of functions available.

Comment

Cancel Save

Figure 2.2: LeanIX - Edit object

Additional documents can be linked to each object by adding an URL and a description. Files cannot be uploaded from a local device but need to be provided through an URL.

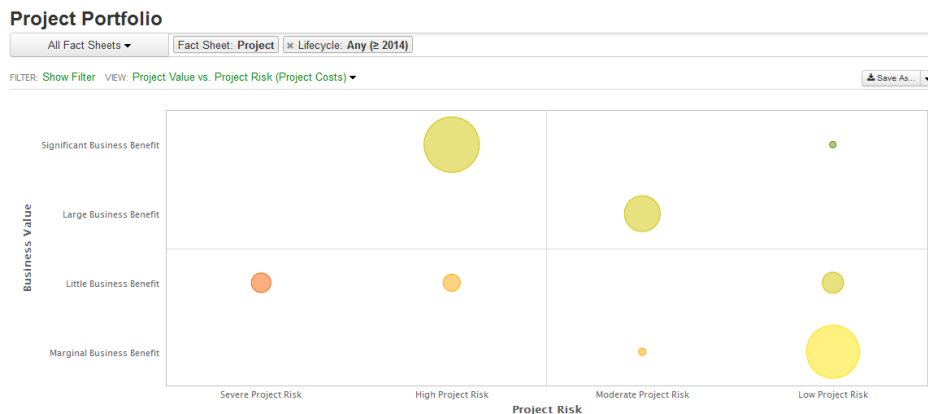


Figure 2.3: LeanIX - Project Portfolio Matrix

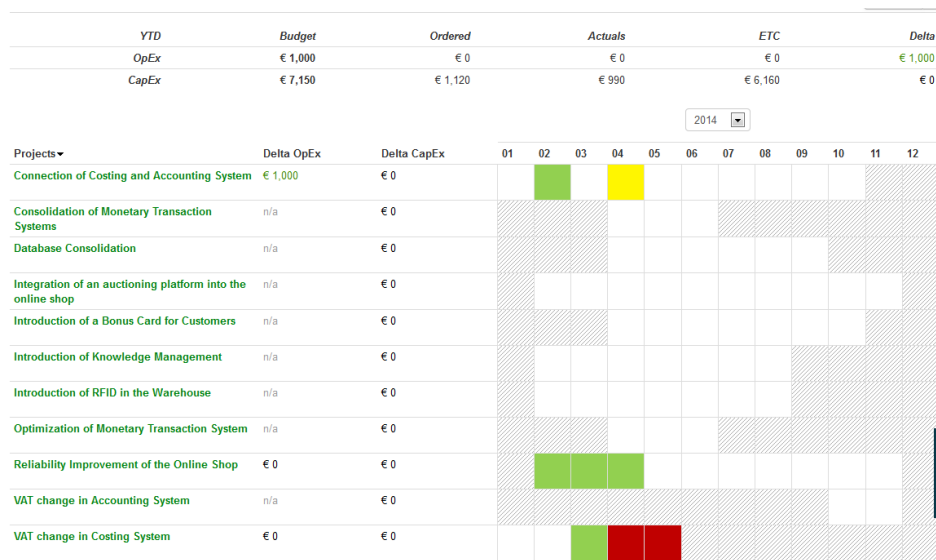


Figure 2.4: LeanIX - Project Cost Matrix

2.1.2 Creating visualizations

There are multiple visualizations existent in leanIX that are grouped into the categories 'Metrics', 'Cost', 'Heat map' and 'Roadmap'.

- The 'Metrics' category offers portfolio matrix visualizations for applications, projects and providers (Fig. 2.3).
- The 'Cost' category offers bar charts for business capabilities, pie charts for provider costs and a project cost matrix (Fig. 2.4).
- In the 'Heat map' category one can generate cluster maps for applications (Fig. 2.5), interfaces and infrastructure as well as location maps for application usage (Fig. 2.6), application sourcing and infrastructure location.

2. LeanIX

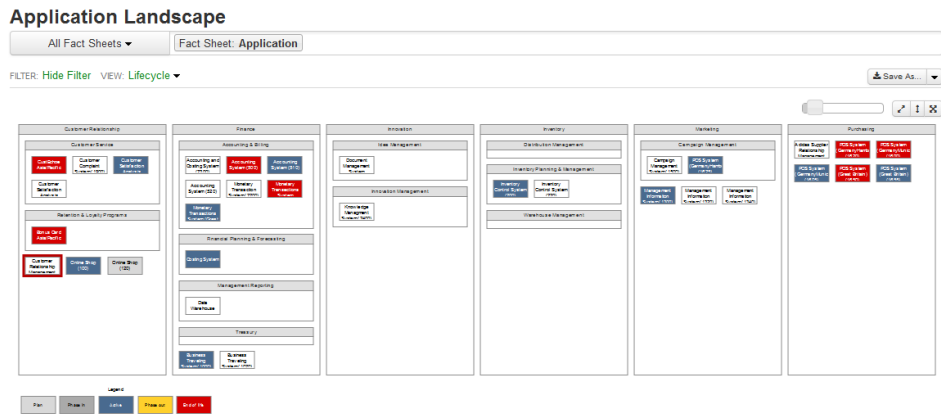


Figure 2.5: LeanIX - Application landscape

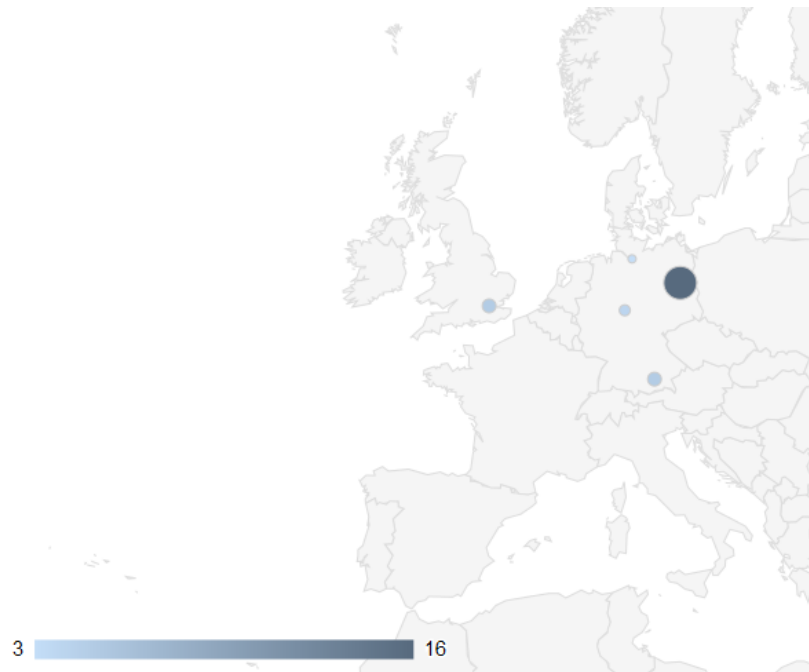


Figure 2.6: LeanIX - Application usage per city

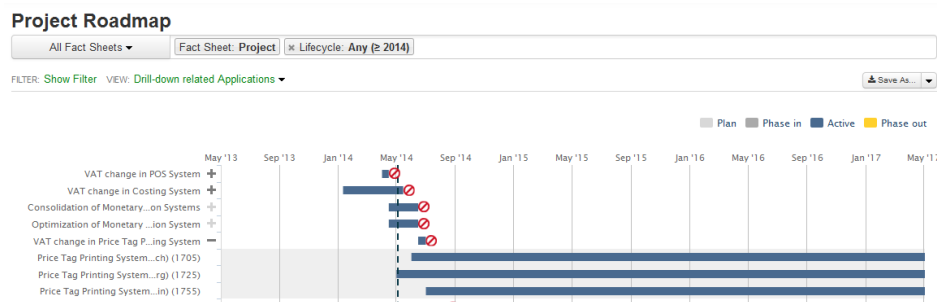


Figure 2.7: LeanIX - Project roadmap

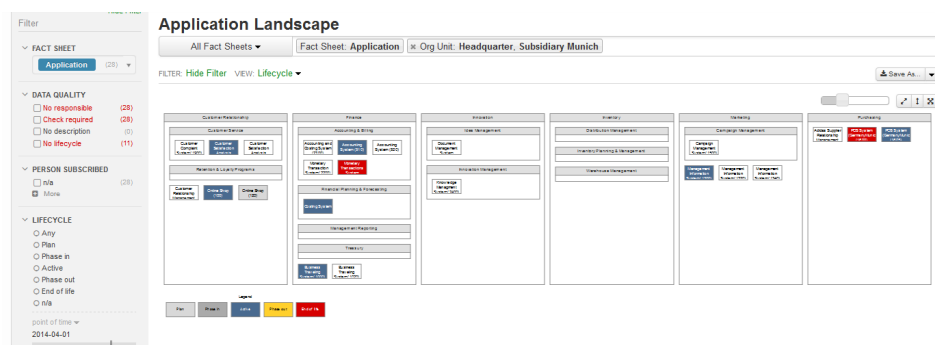


Figure 2.8: LeanIX - Application landscape visualization with activated filter

- Time interval maps are available in the 'Roadmap' category for applications, projects (Fig. 2.7) and infrastructure.

Visualizations are generated automatically and displayed as interactive elements in the browser. The object types to be displayed are predefined per visualization and cannot be changed. Although drawing manual visualizations is not possible, the tool provides filters that are dynamically updated so that an additional query language is not necessary.

2.1.3 Interacting with, Editing of, and Annotating Visualizations

Adaptions of generated visualizations can be made through activating or deactivating filters. Filters can be activated according to attributes like time interval, lifecycle, organizational unit and many more (Fig. 2.8).

Filters are displayed at the top of the visualization and can be deactivated there easily. Moreover, the color coding of objects can be changed in some visualizations. For example, in the application landscape diagram application colors can be adapted based on lifecycle, project status or functional/technical fit.

After regenerating diagrams, earlier applied filters are deactivated. In the evaluated version the only way to save filtered visualizations is to save them as bookmarks in the browser. With the help of a bar one can zoom into visualizations to get a detailed view (Fig. 2.9).

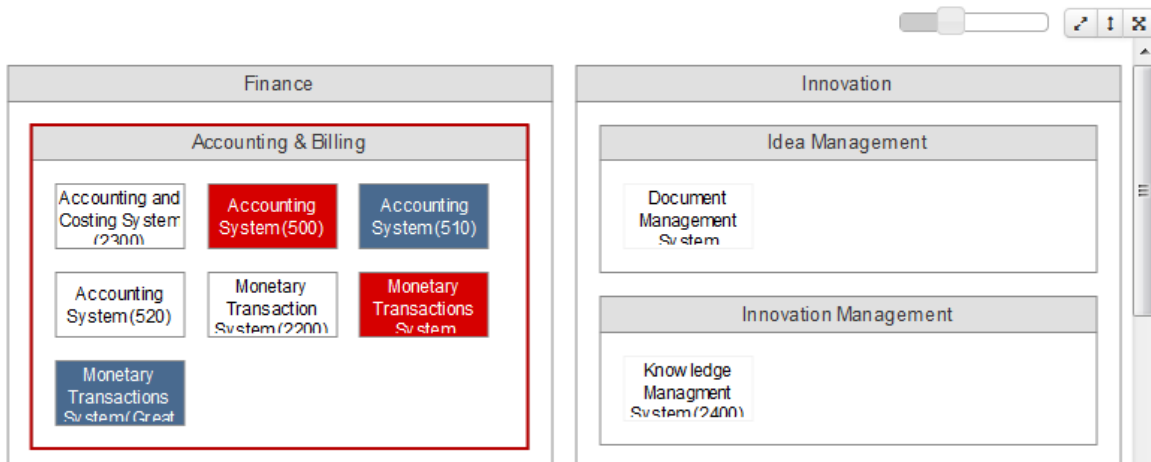


Figure 2.9: LeanIX - Detailed view through zoom bar

Groups of objects can be selected, e.g. applications belonging to the same business domain. A list of these applications is then displayed below the visualization. Through these lists, specific objects, e.g. affected applications, can be accessed directly.

2.1.4 Communication and Collaboration Support

The tool vendor, LeanIX GmbH, states that they put special attention on collaboration while developing their software. There are three kind of user roles one user can get assigned in leanIX: *Viewer*, *Member* and *Admin*. Viewers can view and subscribe to all objects. Members can modify and create new objects, and admins can manage other users and export data into other environments.

Every user can subscribe to objects as *Responsible*, *Observer*, or *Accountable* to get directly informed if any changes are made on the subscribed object. A subscription has the advantage that each user can filter for the objects he or she is interested in and one can directly contact those responsible via e-mail. One can get a quick overview about all subscribed objects. Moreover, visualizations can be adapted that only subscribed objects are included.

It is possible to comment on every object and to open new topics, e. g. users can ask specific questions or provide feedback. Other subscribed users get notified and can reply to this open topic (Fig. 2.10). This feature facilitates discussions and collaboration.

Furthermore, leanIX integrates a tagging service that allows users to structure content according to individual categories. Tags can be assigned to every object instance on-the-fly. In this way objects can be categorized and grouped quickly. It is possible to search for objects with specific tags and use tags to filter visualizations. Objects can be added to existing tag categories through an auto-completion feature. Figure 2.11 shows a snapshot that was taken while tagging an application system.

After logging in, the user gets redirected to a newsfeed where recent changes and updates are displayed (Fig. 2.12). There is a filter possibility to filter for news about subscribed objects,

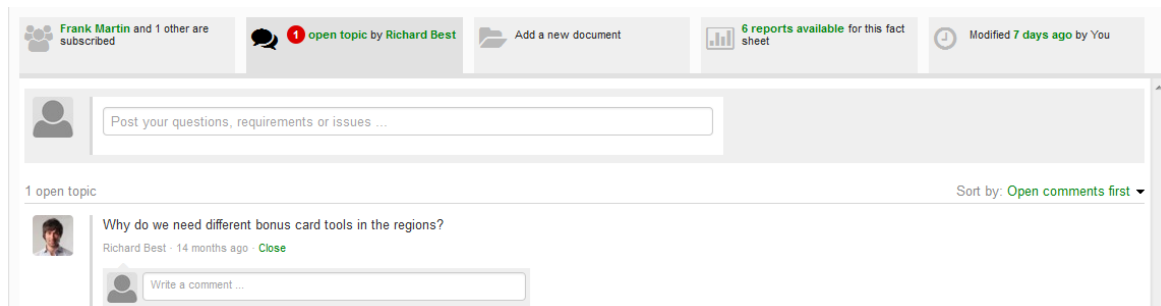


Figure 2.10: LeanIX - Open topics



Figure 2.11: LeanIX - Tagging an application

newly created objects or open topics. Users can subscribe to the newsfeed via RSS feeds to access it from other locations.

2.1.5 Flexibility of the Information Model

LeanIX brings a fixed information model where additional object types cannot be created. The names of existing object types can be changed but semantics remain the same (Fig. 2.13).

New custom fields can be added that can be assigned to object types. Different values can be defined for custom fields. These values can then be linked to object instances by tagging the object instance with a specific value. The resulting tags can be used in visualizations for filtering and color highlighting (Fig. 2.14).

2.1.6 Support of large scale Data

During the evaluation leanIX works smoothly and quick with a small amount of data of around 40 applications and corresponding relations and interfaces. The import of a bigger amount of applications takes a few minutes but is successful. Unfortunately, the mass update feature is very limited and only allows to mass-update few attributes. Important attributes

2. LeanIX

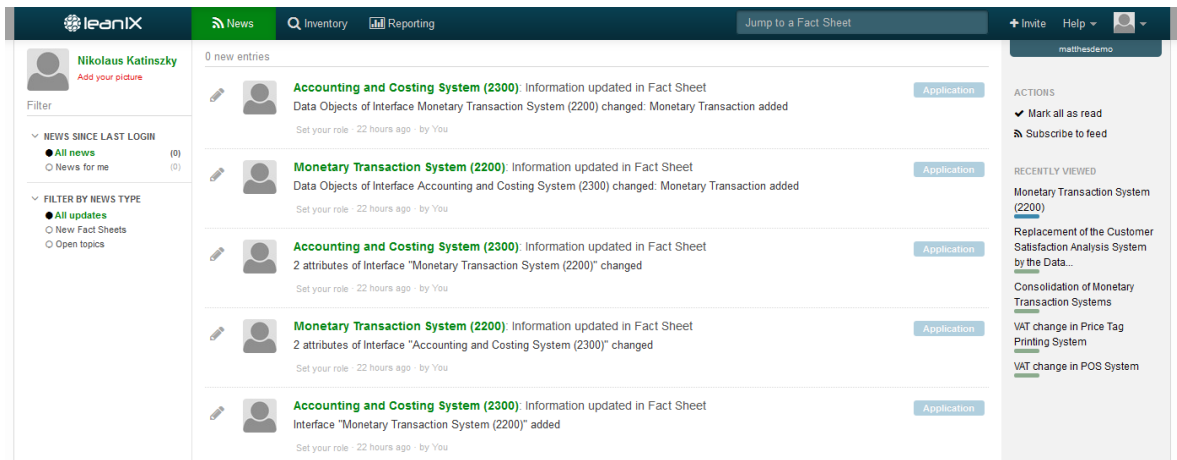


Figure 2.12: LeanIX - Newsfeed

Fact Sheet Definition Customize name and definition of a Fact Sheet

	Description	Quality Check
Bus. Capability	Which Business Capabilities are supported by Applications?	
Process	Which Processes are driving the business?	
User Group	Who is using which Application, how many users are using an Application?	365 Days
Project	Which Projects are currently changing the IT landscape?	
Application	Which Applications are provided by IT to support the business?	90 Days
Data Object	Which Data Objects are created, modified or transmitted via interfaces?	
IT Component	What are underlying software, hardware and services used to operate Applications?	365 Days
Provider	Who is providing or delivering these IT Components?	365 Days
Tech. Stack	How are the IT Components grouped, which skills or technology-groups are required?	

Figure 2.13: LeanIX - Metamodel object types



Figure 2.14: LeanIX - Visualization with highlighted custom values

like interfaces or location need to be updated manually by the user in leanIX which can be very exhausting.

The infinite scrolling implementation of object lists is helpful for large amount of data because only the data that is currently viewed by the user is loaded. If the user scrolls through a list, the list gets extended automatically. In this way, the user does not notice that there is that much data in the system. The search field is also very useful to access one specific object quickly, no matter how many objects exist.

The overview in visualizations is limited with a large amount of data. Either text labels are difficult to read or the user has to scroll a lot manually inside the visualization to get text labels illustrated in a proper size. Performance issues cannot be noticed.

2.1.7 Reporting

There are no tabular reports available in leanIX. The *Reporting* tab in leanIX only includes visualizations. Creating custom reports or visualizations is not possible. Only predefined visualizations where object types cannot be changed are available. Below each visualization there is a list of selected objects. This list links to detailed information.

Generated visualizations can be filtered for different attributes. Some visualizations offer a drill-down functionality. For example, projects can be drilled down to see affected applications. Some visualizations are especially useful for reporting. The project cost tracking visualization (Fig. 2.4) illustrates the status of all projects as well as planned and current costs. Filtered and adapted visualizations can be saved as local browser bookmarks. They can be exported as PDF files.

2.1.8 Usability

LeanIX is easy to use. New users get to know the application very quick. The tool has a clear structure that always provides a good overview. There are only three main menu items where every functionality can be accessed.

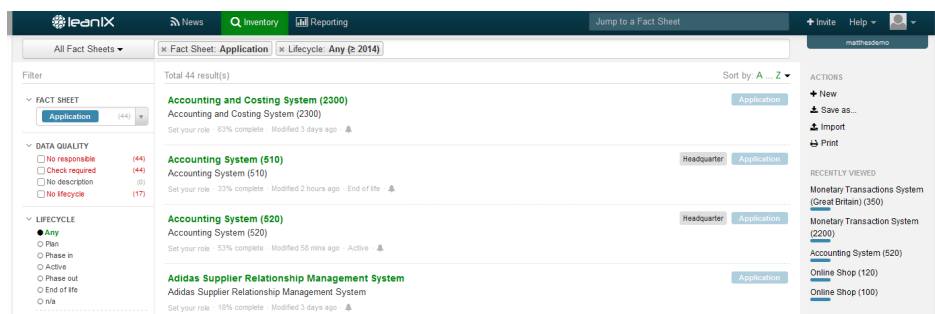


Figure 2.15: LeanIX - Overview

The '*Jump to Fact Sheet*' field with search and drop down selection is especially useful to access objects quickly. The newsfeed is helpful for the user to get a quick overview about all changes and events. The tagging functionality allows quick grouping.

The infinite scrolling implementation where additional content at the bottom of the page is loaded automatically creates a modern, lean look and feel.

The leanIX documentation is accessed through a web browser. The level of detail is limited but the most important questions are answered.

2.2 Evaluation of EA Management Support

In this section leanIX's EA Management support capabilities are evaluated.

2.2.1 Landscape Management

LeanIX supports landscape management through predefined visualizations and application attributes. The application landscape diagram (Fig. 2.5) allows to filter for different points in time. The application attribute 'Lifecycle' can be highlighted according to one's application status: 'Plan', 'Phase in', 'Active', 'Phase out' or 'End of life'. The value 'target' in order to define a special target landscape is not available but could be realized through assigning an application lifecycle.

The visualization 'Interface map' (Fig. 2.16) is a cluster map that represents relations between applications. The interface type can be highlighted through color coding. Color coding can be applied for other relevant application attributes like business criticality, functional or technical fit, project status or technology risk.

The visualization 'Application roadmap' (Fig. 2.17) is useful to see when applications were or will be exchanged by other applications. There is a drill down functionality to see successor applications, child applications, related projects or IT components.

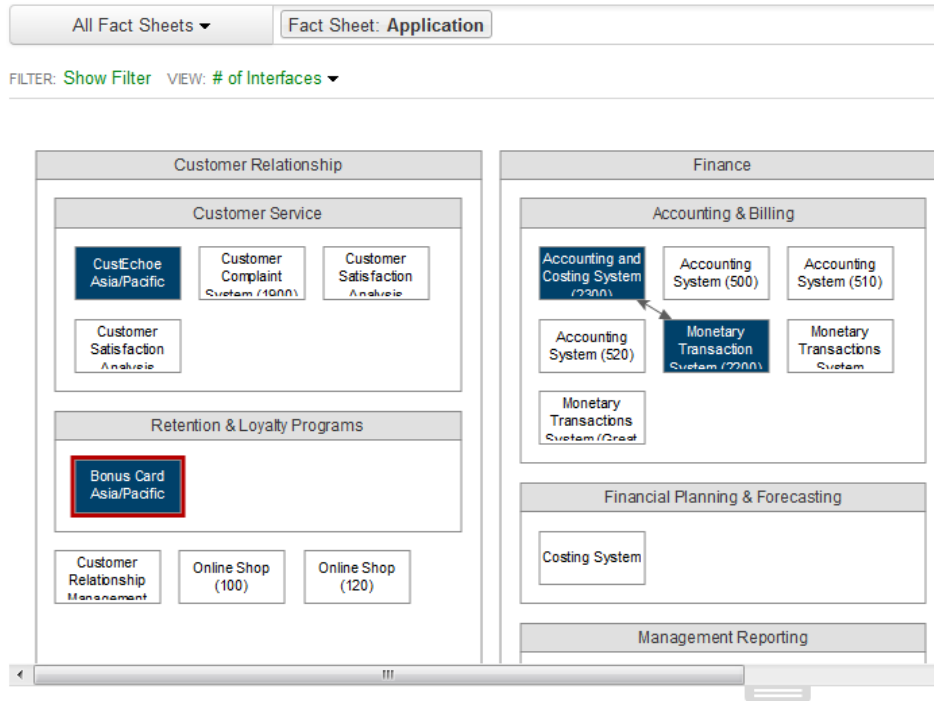
The given filtering functionality in all visualizations can also be useful for landscape management. In this way, applications can be filtered for specific criteria like business processes, used IT components, lifecycle, manually assigned tags and other attributes.

2.2.2 Demand Management

There is no object type demand or requirement integrated in leanIX. The manual tagging functionality could be used to reflect demands. Applications or projects could then be tagged with a demand the corresponding objects try to meet. These manually assigned demands can also be highlighted through color coding in visualizations. Another option is that demands are expressed via comments through the *Open topic* functionality.

Applications include an attribute called 'Functional fit' that reflects how well one application meets business requirements. A number between 1 and 4 can be assigned to this attribute.

Interface Map



ATTENTION: 15 entries not included in this report due to missing data. [Show list](#)

List of Interfaces

Total 1 result(s)

Accounting and Costing System (2300)	→	Monetary Transaction System (2200)	Weekly	Batch
--------------------------------------	---	------------------------------------	--------	-------

Figure 2.16: LeanIX - Interface map

Application Roadmap

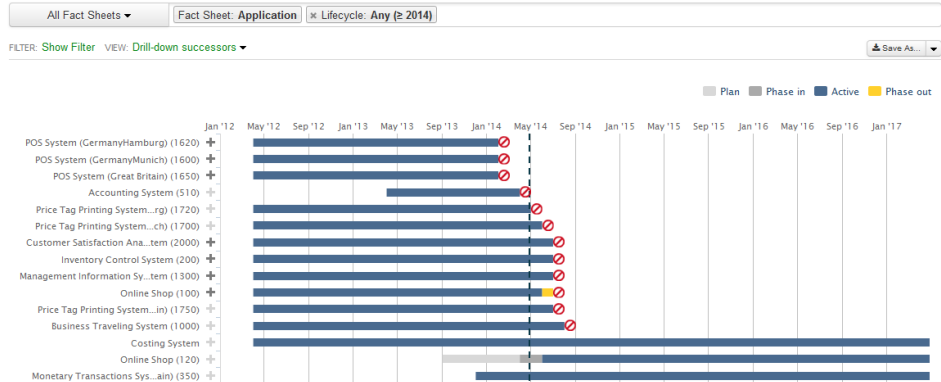


Figure 2.17: LeanIX - Application roadmap

2.2.3 Project Portfolio Management

There is an object type 'Project' integrated into leanIX' information model. This object type has the following predefined attributes: lifecycle, project value, project risk, affected applications, budget and actual costs, involved providers and their corresponding costs. Moreover, a project status can be updated: it can be set to green, yellow or red, depending on the project situation. Current costs as well as a cost estimation can be entered. These maintained attributes can then be used in visualizations to perform project portfolio management inside leanIX.

The project portfolio matrix (Fig. 2.3) helps to compare projects with each other based on project value, project risk and costs. It is useful to select between projects that shall be performed and others that better be dropped.

The project cost matrix (Fig. 2.4) is useful to track the project status over time as well as planned and actual costs.

A project roadmap visualization (Fig. 2.18) illustrates projects on a time interval map. Successor projects or related applications can be integrated through the drill down functionality.

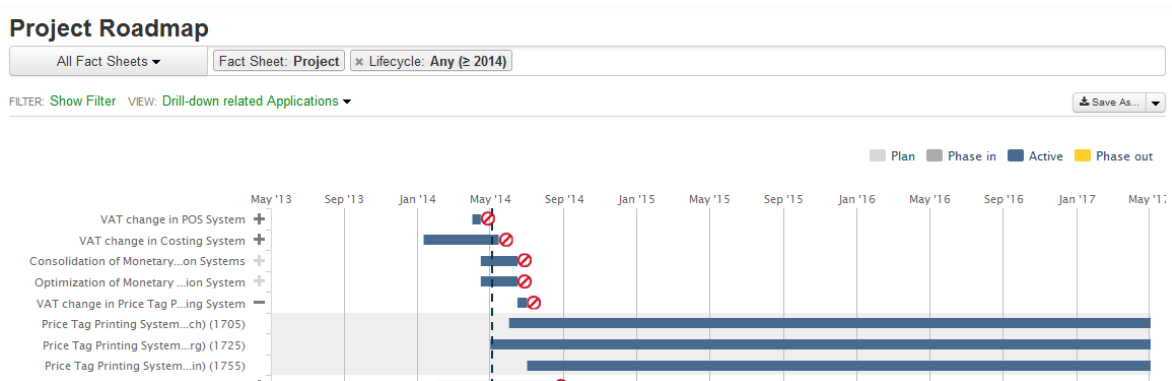


Figure 2.18: LeanIX - Project roadmap

Filtering options are available in all visualizations in order to filter for specific project attributes, affected applications providers or manually assigned tags. All these features and visualization options are helpful for performing typical project portfolio management tasks like project selection, time and cost tracking or the identification of dependencies.

2.2.4 Synchronization Management

Synchronization management can be performed in leanIX by using the visualization project roadmap (Fig. 2.18). Applications that are affected through projects can be visualized. Manual annotations or adaptations are not possible. Business units or other EA elements that are affected through particular projects cannot be mentioned in the project detail page. There are only relations between projects and applications available. Successor and predecessor projects can be named. These connections do not consider project delays.

2.2.5 Strategies and Goals Management

The object types 'Strategy' or 'Goal' are not available in leanIX. LeanIX focuses on IT-relevant areas. Some object type attributes can be used for strategic management: the application criticality can be useful for strategic decisions as well as project value, project risk and the corresponding project portfolio matrix (Fig. 2.3) may be relevant. Another interesting part for the IT-Strategy could be the object type 'Provider' where hardware and software providers can be integrated. There is an interactive pie chart visualization available that compares the provider costs (Fig. 2.19).

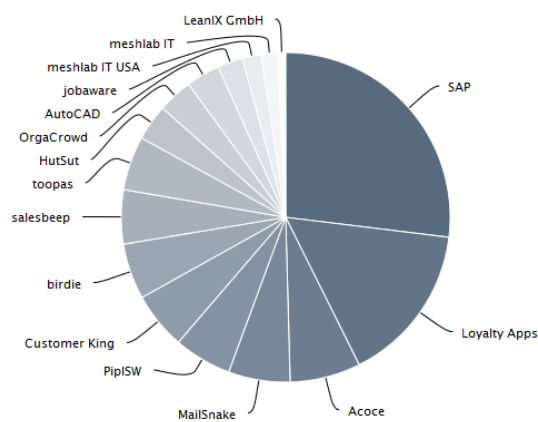


Figure 2.19: LeanIX - Provider cost visualization

2.2.6 Business Object Management

Business object management is supported in leanIX through the object type 'Data Object'. Applications and operations that are performed on these objects can be set (Fig. 2.20).. Moreover, the interface that the business object is transferred through can be specified. The interface direction, frequency, type and technology can be named.

Visualization options are limited. A filter can be set that only applications are displayed that use particular data objects inside the application landscape visualization and inside the interface map. The actual data object itself is not visually represented.

2.2.7 SOA Transformation

LeanIX does not offer any business or application service object types. An interface map can be used to identify potential applications that could be replaced with services. Filter options and the application roadmap (Fig. 2.17) can be used to see the application development over time. The application landscape diagram (Fig. 2.5) illustrates which applications support which business capabilities. The filter can be used to focus on specific business processes, locations or other relevant criteria. There are no layered views or reports available in leanIX.

2. LeanIX

Information		25% complete
NAME & DESCRIPTION	Customer Complaint	
LIFECYCLES	What is the lifecycle state (e.g. planned, active or end of life)?	<input type="button" value="Edit"/>
SUCCESSOR (Optional)	What are the successors after end-of-life?	
PREDECESSOR (Optional)	What are the predecessors?	
Data Management		33% complete
CLASSIFICATION	What is the security classification of this Data Object?	
APPLICATIONS	Which Applications are creating, modifying or deleting this Data Object?	
	Customer Complaint System (1900)	Usage: Create, Update, Delete
	Customer Relationship Management System (2100)	Usage: Create, Update, Delete
INTERFACES	By which interfaces is this Data Object transferred?	

Figure 2.20: LeanIX - Data object details

2.2.8 IT Architecture Management

Architectural blueprints can be managed in leanIX through the predefined object type 'Technical Stack'. Children or parent objects can be linked to build structures. Blueprints that use same architectural elements are difficult to model because one element can only be linked with one parent object. IT components can be assigned to technical stacks. In this way, it is possible to look up by which technical parts architectural blueprints are realized.

The IT component diagram (Fig. 2.21) can be used to see which IT components rely on which blueprints. The visualization can be filtered to focus on specific blueprints or technologies. A time dimension filter is useful to see the development over time. The lifecycle attribute can be highlighted through color coding.

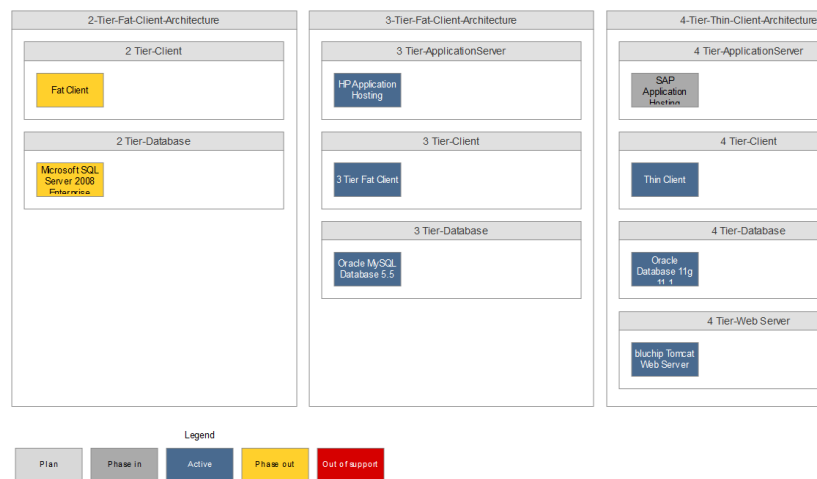


Figure 2.21: LeanIX - IT component diagram

Application objects can use IT components. Through this relation it is possible to look up which blueprint they follow.

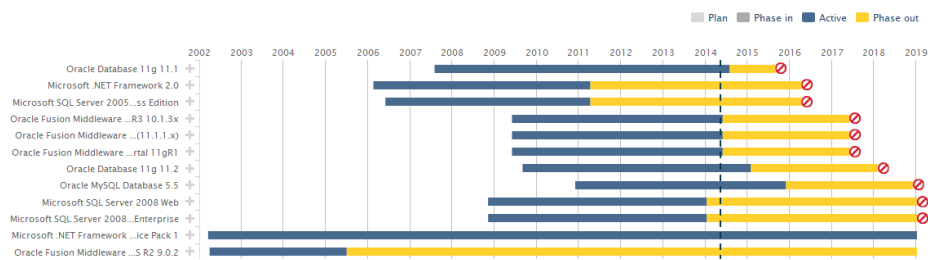


Figure 2.22: LeanIX - IT component roadmap

2.2.9 Infrastructure Management

IT components in leanIX provide different attributes that support infrastructure management. They can be linked to a technical stack which can be regarded as an architectural blueprint. It can be stated which applications use the IT component. The attribute 'Technical fit' reflects the IT component's shape from a technical view. Moreover, a provider, lifecycle status and location can be set.

The provider cost pie chart (Fig. 2.19) can be used to see cost that infrastructure providers cause. The IT component roadmap (Fig. 2.22) is helpful see development over time. It can be analyzed when one component shall be replaced or when license support ends. Moreover, affected applications can be integrated.

SAP Sybase PowerDesigner

The SAP Sybase PowerDesigner is a business process and data modeling tool supporting the design of various types of EA models including process maps, application architecture diagrams, and technology infrastructure diagrams. A full list of supported model types can be seen in Fig. 3.1.

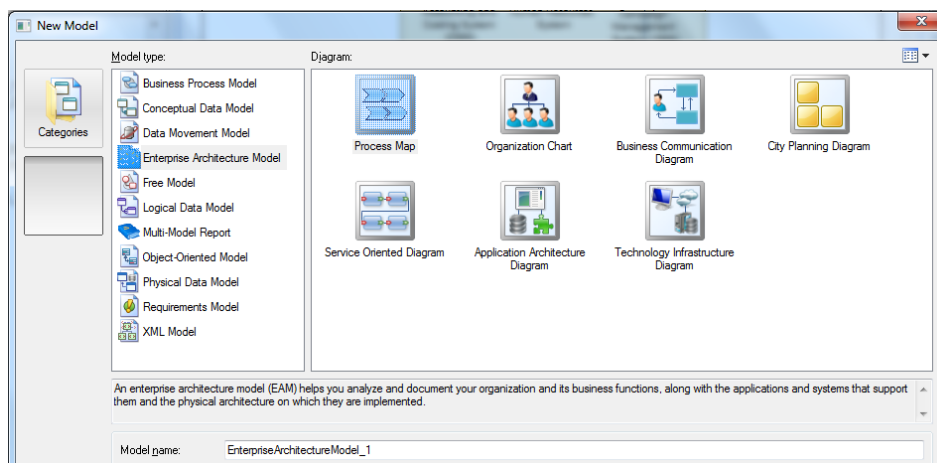


Figure 3.1: PowerDesigner provides multiple model types for designing EAs

In our evaluation we used the free online trial version 16.5 that can be tested for 15 days¹. The trial version offers all features of the purchasable version for a limited amount of time. In our evaluation the Enterprise Architect module was mainly used to evaluate EAM specific features.

¹A free trial version can be downloaded here (registration required): <http://www.sap.com/pc/tech/database/software/powerdesigner-enterprise-architect/index.html>

3. SAP Sybase PowerDesigner

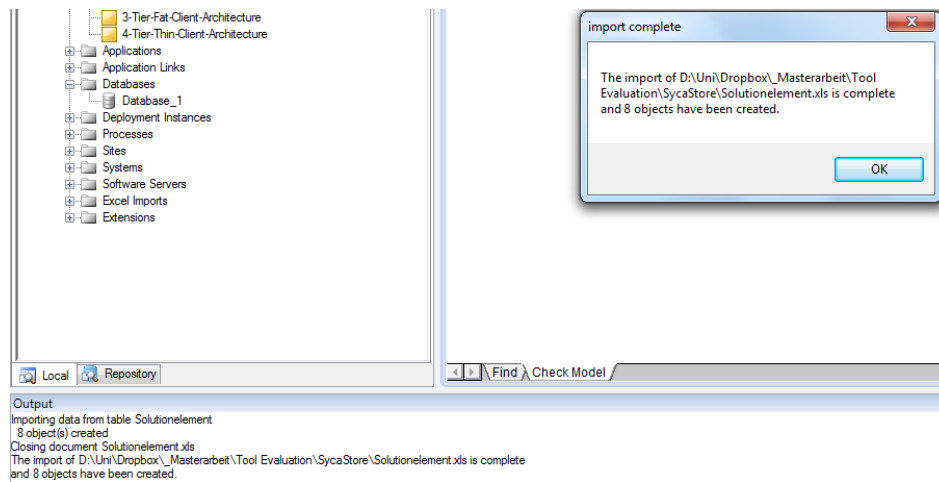


Figure 3.2: After completion of the import, the data is available through the object browser on the left. The output panel at the bottom shows messages from the import process.

3.1 Evaluation of Specific Functionality

This section describes the results of the scenario simulation for specific functionality.

3.1.1 Importing, Editing and Validating

In PowerDesigner, data objects like applications can be imported from one or multiple Microsoft Excel spreadsheets. An imported Excel spreadsheet may contain multiple worksheets, whereas each worksheet represents different kinds of objects. For each column in the Excel spreadsheet the user can decide whether to import the column as an attribute of the imported object, or as a list of associated objects. Moreover, columns can either be mapped to existing predefined attributes of PowerDesigner objects, or to new custom attributes. Fig. 3.2 shows a successful import of an Excel spreadsheet. Once the import is successfully completed, the data objects are available through an object browser and, e.g., can be used for the creation of new diagrams.

When importing an Excel spreadsheet to PowerDesigner, a validation mechanism checks whether the imported objects already exist. However, a detailed report of the validation is not presented. In case of an error during the import process, respective messages are shown in the output window and the import process needs to be restarted.

Objects in PowerDesigner are editable by adding new or changing existing attributes. The corresponding form is shown in Fig. 3.3. The set of predefined attributes depends on the actual type of the object, e.g., an application object has the attributes *license number*, *cost*, *expiration date*, *programming language* and *operating system*. In addition to the predefined attributes, PowerDesigner also allows the creation of new attributes for a certain object type. Thereby, the definition of an attribute includes the attribute type as well as one or multiple default values.

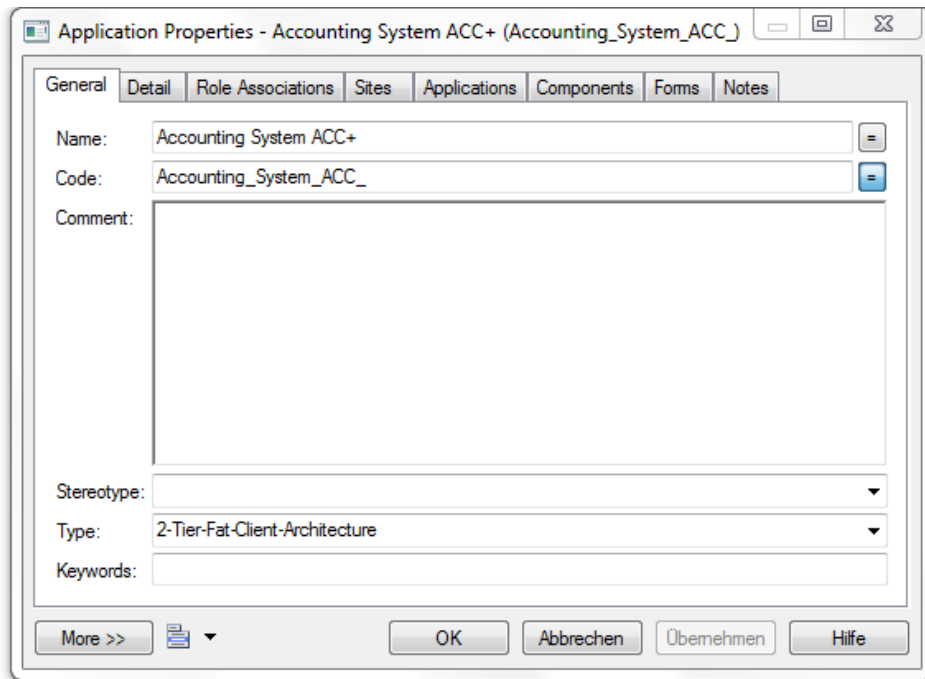


Figure 3.3: PowerDesigner - Editing properties of an object

PowerDesigner does not support the export of its data objects through a common file format. Instead the data objects can be saved as proprietary .eam files, which in turn are XML files adhering to a certain schema. These .eam files can be imported with PowerDesigner. Furthermore the objects of PowerDesigner's EA module can be exported to low-level PowerDesigner modules.

3.1.2 Creating visualizations

PowerDesigner's Enterprise Architect module offers three categories of visualizations: Business diagrams, application diagrams and technology diagrams. A diagram category determines the set of objects which can be used in a graphical representation, e.g., an application diagram does not contain any business related object types.

The business diagrams category includes the following visualizations:

- An organization chart visualizes the structure of an organization.
- A business communication diagram illustrates business flows and relationships between organizational units, business functions, roles and sites (Fig. 3.4).
- A process map can be used to model the order of connected business processes (Fig. 3.5), whereas processes can be grouped into business functions. However, sub-processes cannot be modeled.

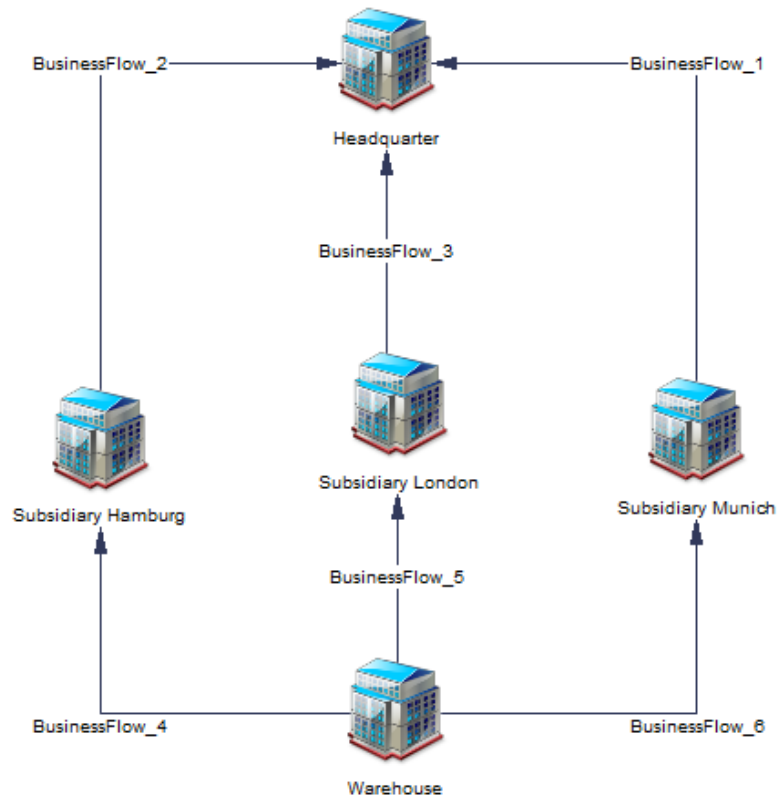


Figure 3.4: PowerDesigner - Business communication diagram

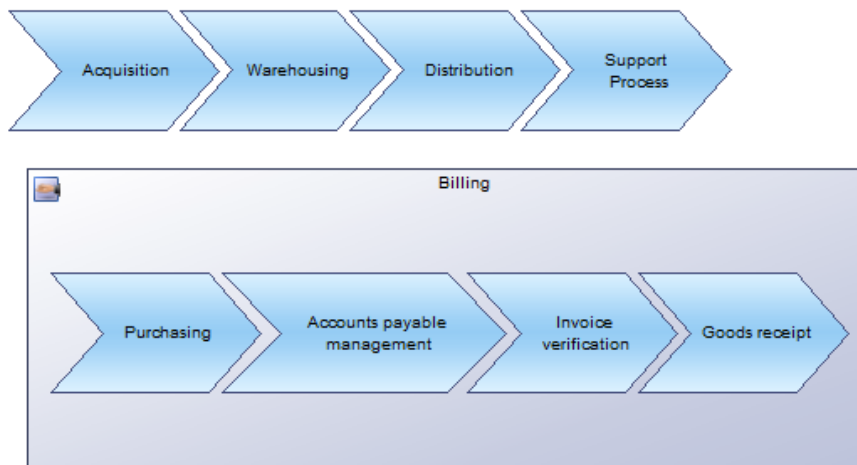


Figure 3.5: PowerDesigner - Process map

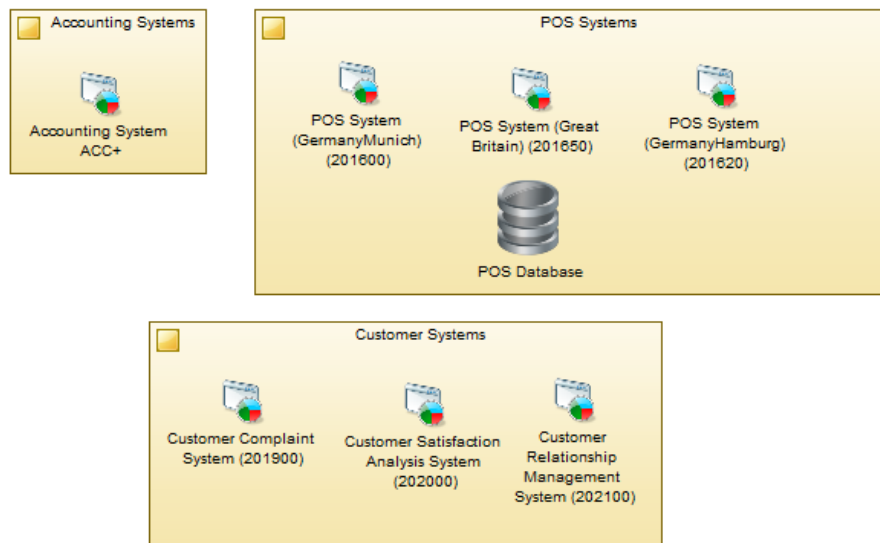


Figure 3.6: PowerDesigner - City planning diagram

- A city planning diagram represents a big picture of an application landscape. It can be used to group applications and infrastructures according to their business needs (Fig. 3.6).

In addition to business diagrams, PowerDesigner also provides two diagrams focusing the application perspective of an EA:

- An application architecture diagram can be used to model where certain systems, applications and services are deployed, and which of those elements are related to each other. The level of detail is up to the modeler who is drawing the diagram. Fig. 3.7 shows an overview of an application architecture, whereas Fig. 3.8 shows a more detailed view.
- A service oriented diagram can support the SOA design phases by visualizing the relationships between business, application services and the technology layer (Fig. 3.9).

The only visualization of the technology diagram category is the technology infrastructure diagram,

It has to be noted that PowerDesigner has a strong focus on modeling. As a consequence, all diagrams need to be created manually, i.e., there is no automated generation mechanism. To add a certain object to the diagram, it can be included to the diagram by simply drag&drop the objects from the object browser onto the modeling area.

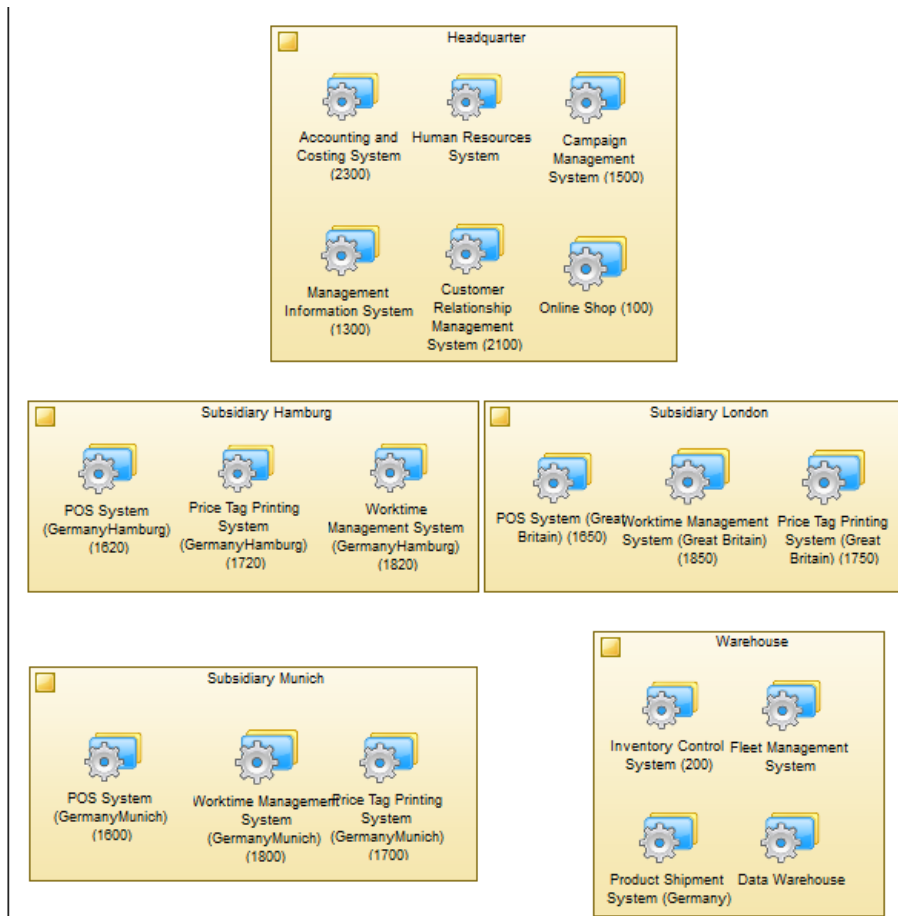


Figure 3.7: PowerDesigner - Application architecture diagram overview

3.1.3 Interacting with, Editing of, and Annotating Visualizations

PowerDesigner Enterprise Architect supports interacting and editing diagrams, i.e., objects can be formatted by changing colors, graphical shape, font or adding a shadow. For example, Fig. 3.10 shows how to customize an object's filling by changing colors and image gradients.

Moreover, the graphical icons representing an application or infrastructure element can be replaced with individual graphics or icons. They can be aligned either top, horizontal central, vertically, etc.

Annotations to diagrams can be made in any diagram category by adding so-called drawing tools (e.g., labels, lines, and other graphical shapes) from PowerDesigner's toolbox (Fig. 3.11). The filling, line color, font and other attributes of this drawing tools are customizable as well.

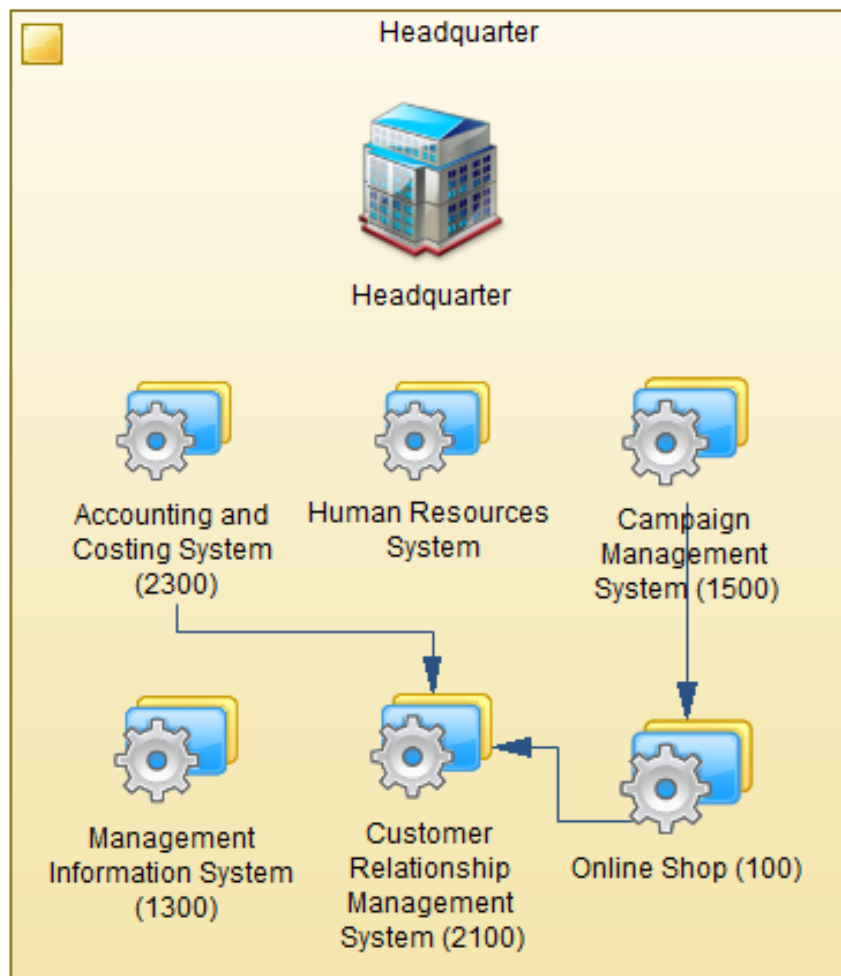


Figure 3.8: PowerDesigner - Application architecture diagram detailed view

3.1.4 Communication and Collaboration Support

Models in PowerDesigner can be saved as proprietary files and subsequently imported by other users on other PC devices. In this way multiple users are able to read and edit the same EAM model. To improve collaboration, PowerDesigner integrates a repository, which can be put into a network and acts similar to a database. Multiple users can connect to it with a user/password combination and log into the repository simultaneously. The repository can store both full models and objects. Moreover, PowerDesigner also supports a check in /check out functionality for repositories (Fig. 3.12).

Communication and interaction features like a user chat could not be found and observed.

3. SAP Sybase PowerDesigner

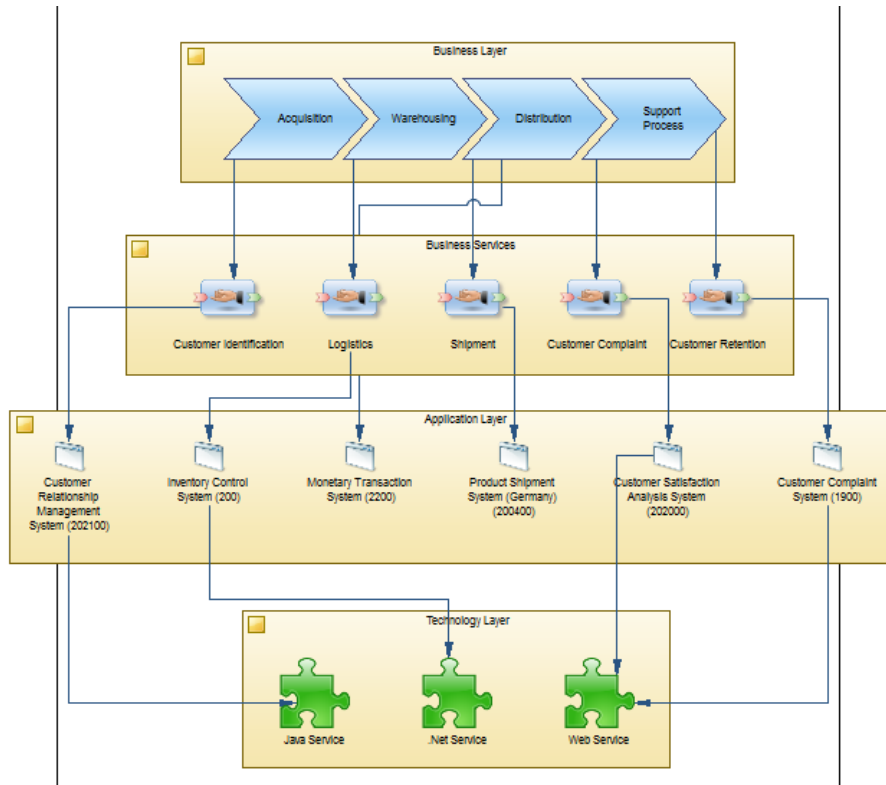


Figure 3.9: PowerDesigner - Service oriented diagram

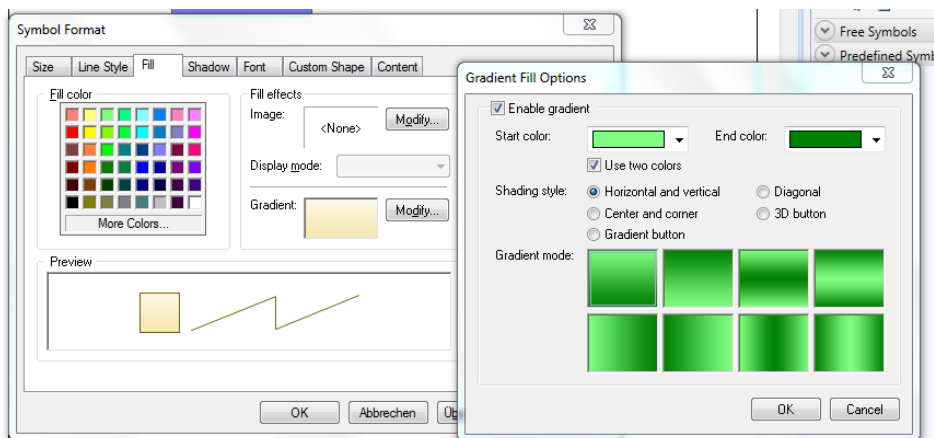


Figure 3.10: PowerDesigner - Edit filling of an object

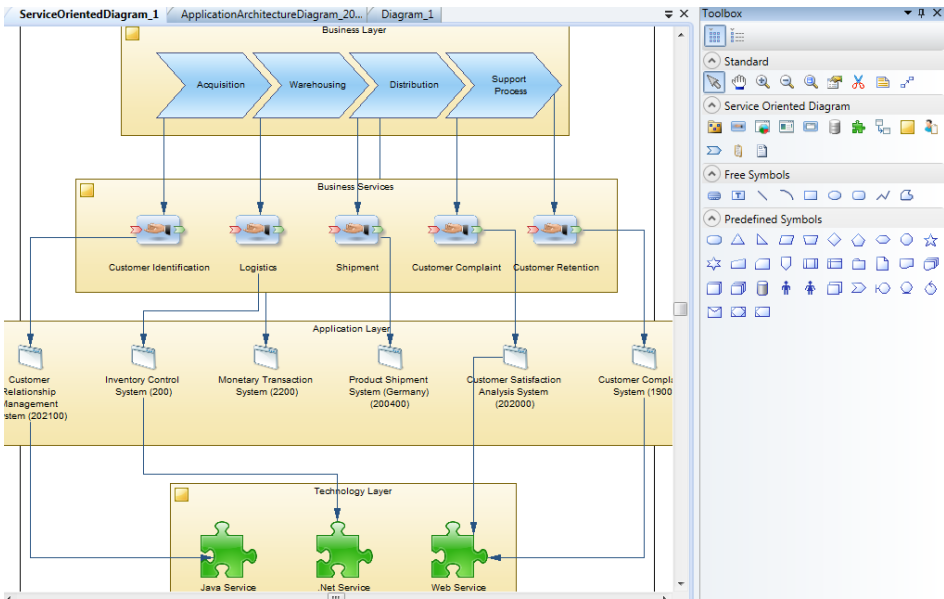


Figure 3.11: PowerDesigner - Available drawing tools

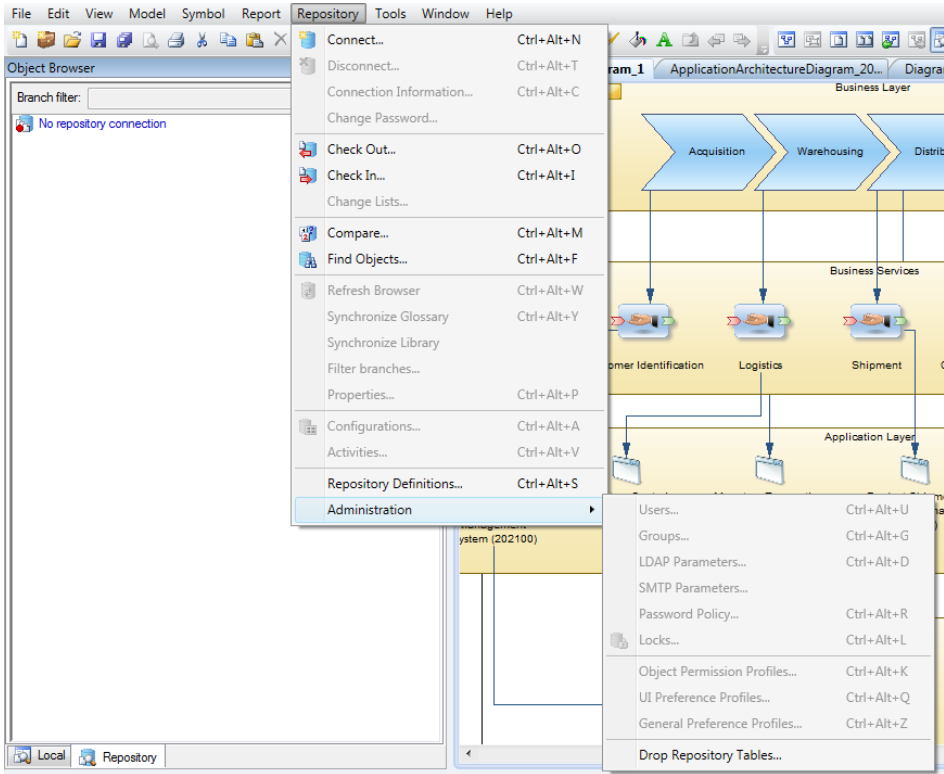


Figure 3.12: PowerDesigner - Repository features

3.1.5 Flexibility of the Information Model

PowerDesigner has neither a fixed predefined meta model, nor does its objects implement any underlying logic. However, there are constraints determining which objects may be used in which diagram types.

As shown in Fig. 3.13, PowerDesigner provides a set of predefined object types. Moreover, additional object types can be created by importing .xem files. While object types already contain a certain set of default attributes, additional ones can be added manually, whereas the user can define the attribute's data type, default value and list of possible values (Fig. 3.14). Relations between objects (e.g., application links) can be defined by designing them in PowerDesigner's modeling area or by creating new objects of the corresponding type.

3.1.6 Support of Large Scale Data

The import process for large scale data does not differ from the aforementioned import process (c.f. Section 3.1.1). However, when containing large scale data, navigating through the object browser becomes difficult and impractical.

In particular the creation of visualizations for large scale data sets is cumbersome, since visualizations have to be drawn manually by either selecting objects from the object browser or creating new objects inside the visualization.

3.1.7 Reporting

In PowerDesigner HTML and RTF reports can be generated automatically. The generated reports illustrate former created models and objects. Fig. 3.15 shows an exemplary HTML report of an application architecture model.

When generating a report, the user can explicitly select the content which has to be included by using a setup wizard (Fig. 3.16). Once the report is generated, the user can modify it by changing the order, captions, displayed content and the HTML/CSS template the report is based on. Moreover, reports can be saved as files and re-generated.

Another type of reporting supported by PowerDesigner is the generation of a list of objects of a certain type, e.g., a list of applications or business processes (Fig. 3.17).

3.1.8 Usability

For the purpose of this tool survey we used the Windows fat client version of PowerDesigner 16.5. All files and data were saved locally.

The drag & drop feature facilitates intuitive modeling and editing of diagrams. The feature to align objects is useful to structure diagrams and give them a proper look. However, the auto-layout mechanism for positioning objects automatically does not convince.

As described in Section 3.1.6, browsing through and in particular visualizing large scale data is cumbersome. On the one hand, each object has to be modeled individually. On the other

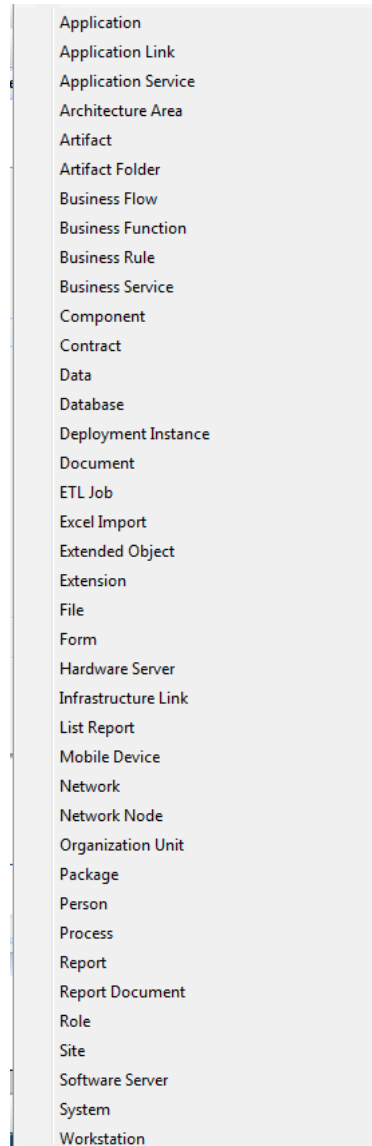


Figure 3.13: PowerDesigner - Available object types

3. SAP Sybase PowerDesigner

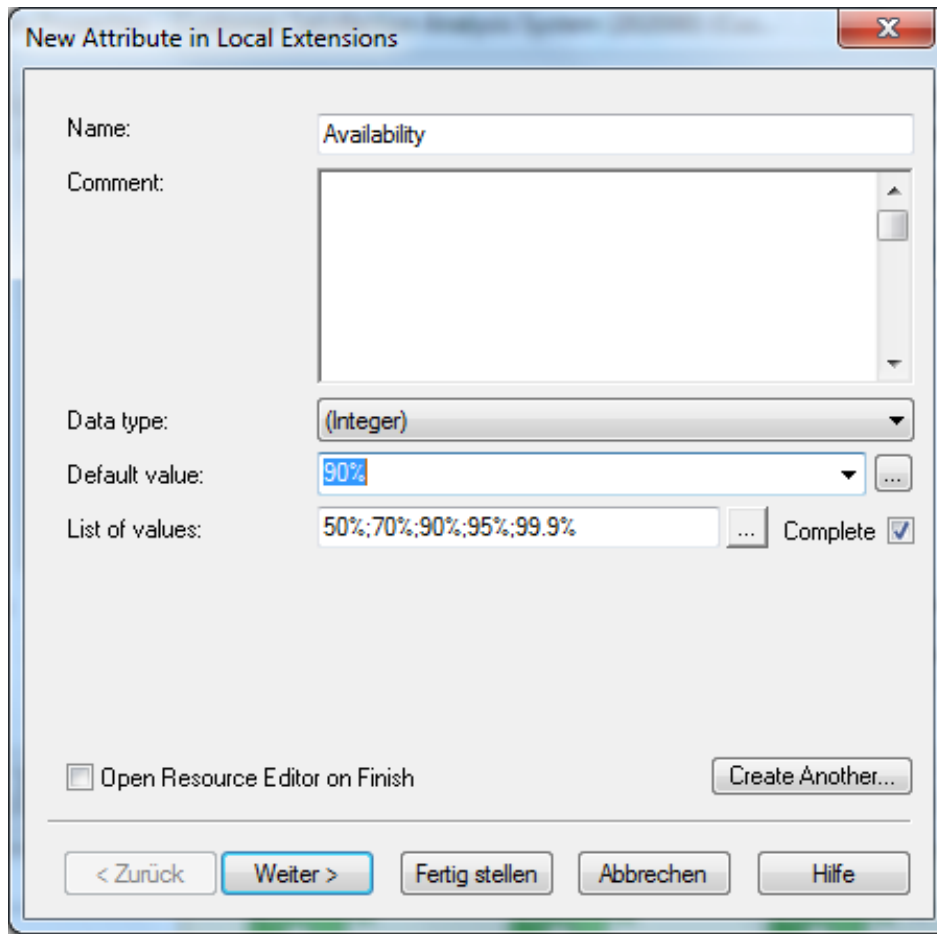


Figure 3.14: PowerDesigner - Creating Attribute

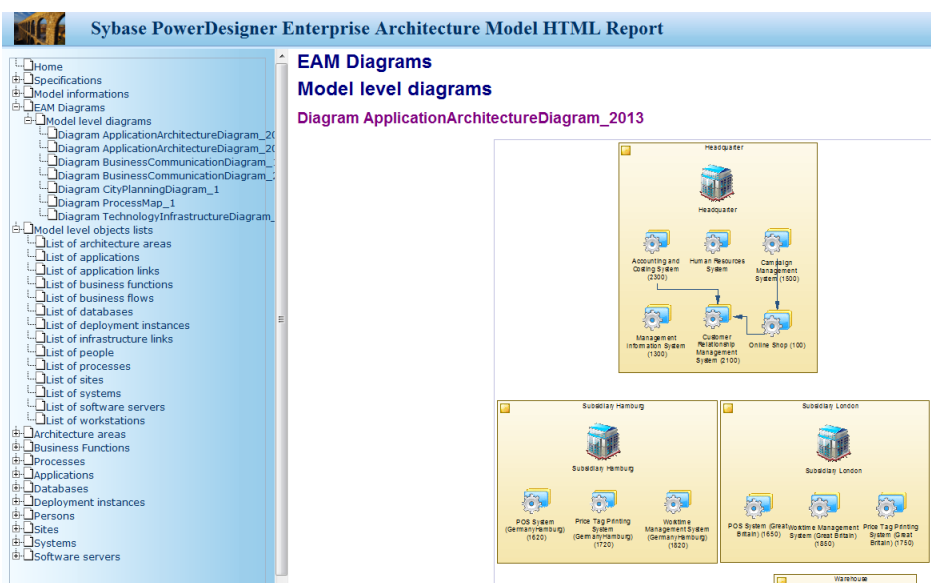


Figure 3.15: PowerDesigner - HTML report of an application architecture model

3. SAP Sybase PowerDesigner

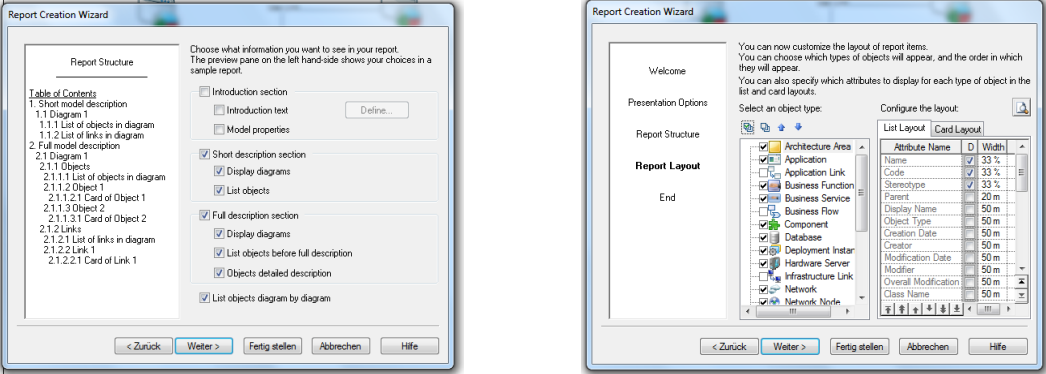


Figure 3.16: PowerDesigner - Step 1 and step 2 of the report generation

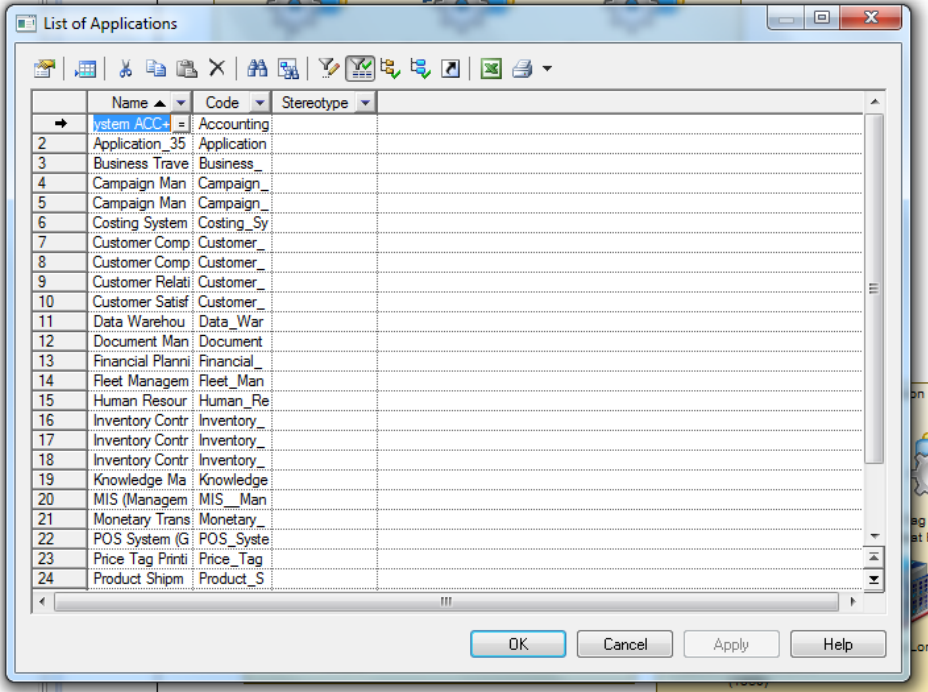


Figure 3.17: PowerDesigner - List of applications

hand, when zooming out of the modeling area to get an overview of the modeled diagram, the symbols and icons become unrecognizable.

Finally, PowerDesigner's Enterprise Architect documentation is mostly about how different diagram types can be modeled manually.

3.2 Evaluation of EA Management Support

In this section PowerDesigner's EA Management support capabilities are evaluated.

3.2.1 Landscape Management

The application architecture diagram type (Fig. 3.8) in PowerDesigner can be used for landscape management. However, the temporal evolution of the landscape cannot be visualized by a single diagram, but has to be done by creating multiple diagrams. Several manually created diagrams can describe different planning scenarios, whereas those cannot be compared directly. Furthermore, differences between scenarios cannot be highlighted automatically, but just manually by adding annotations.

While not supporting the comparison of planning scenarios, PowerDesigner supports the comparison of models. Thereby two models are displayed side by side, whereas differences between these models are highlighted visually. For example, objects and properties that exist in model 1 but are missing in model 2 are colored grey (Fig. 3.18).

An important feature for landscape management is PowerDesigner's support of adding and editing attributes of applications (Fig. 3.14). Predefined application attributes are version, expiration date, cost, location and user roles. Additional custom attributes (e.g., status and productive time period) can be created easily. Furthermore, application system hierarchies can be modeled in the object browser (Fig. 3.19). However, the object browser's hierarchy has no impact onto graphical representations, e.g., by automatically generating nested boxes or relations. Again, this has to be done manually.

3.2.2 Demand Management

The PowerDesigner Enterprise Architect module does not support demand management. However, there is a PowerDesigner Requirements Modeling module for managing lists of demands.

3.2.3 Project Portfolio Management

Since the PowerDesigner Enterprise Architect module does not provide a predefined object type for projects, it is not supporting project portfolio management. Although the creation of a custom object type for projects is possible, PowerDesigner would still lack integration of projects into existing models.

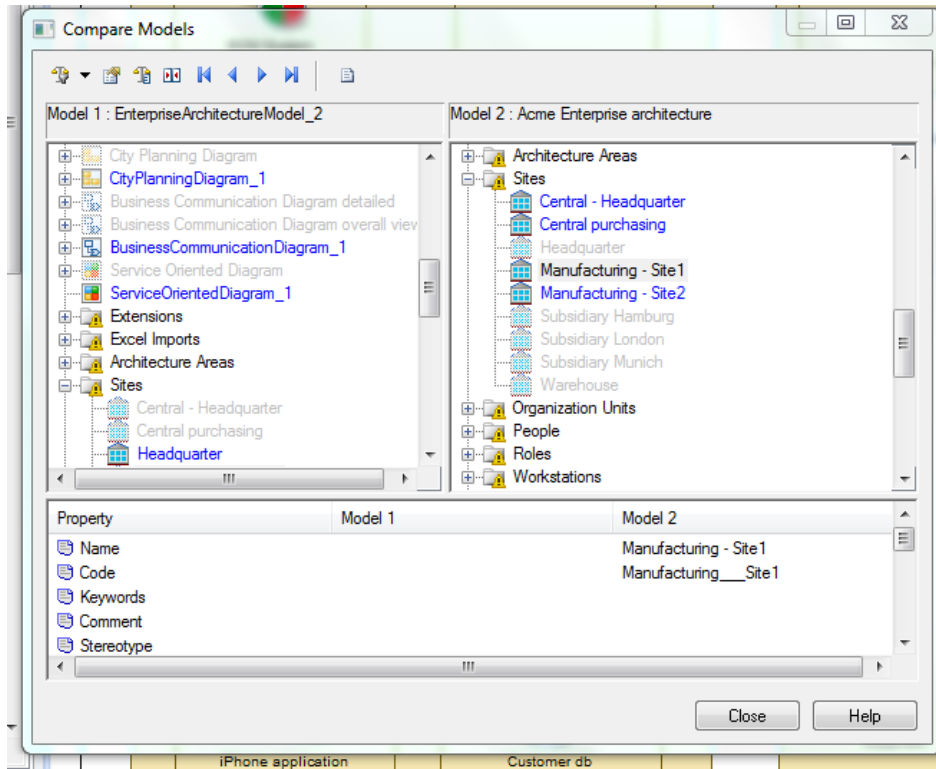


Figure 3.18: PowerDesigner - Model comparison

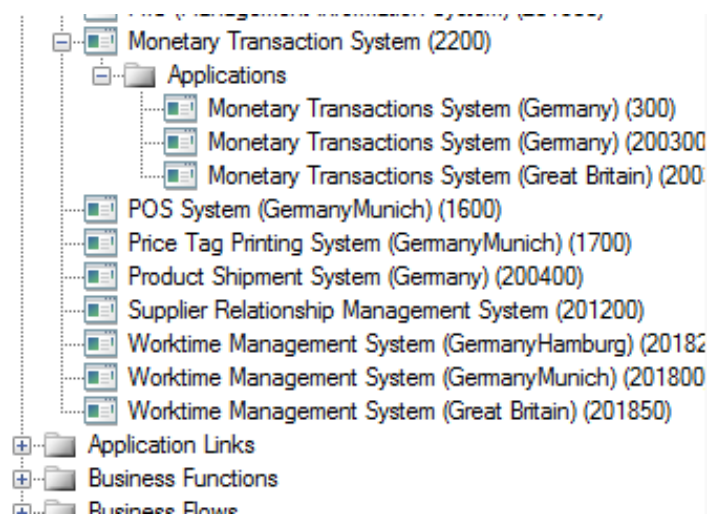


Figure 3.19: PowerDesigner - Application hierarchy

3. SAP Sybase PowerDesigner

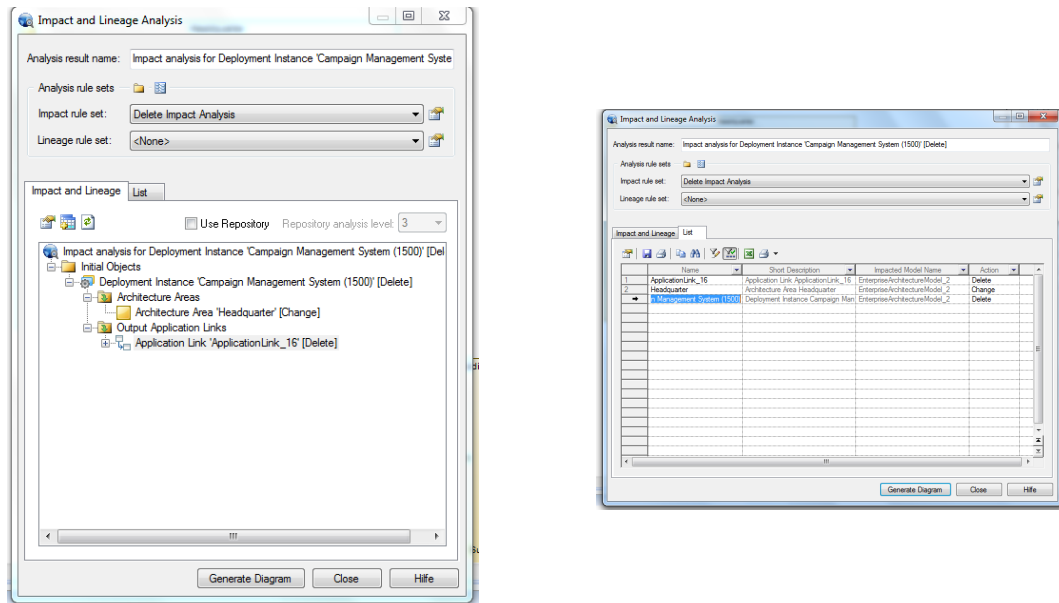


Figure 3.20: PowerDesigner - Impact analysis drop down menu and table representation

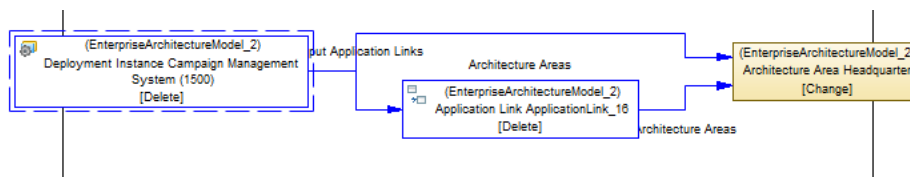


Figure 3.21: PowerDesigner - Impact analysis diagram

However, as described in Section 3.2.1, the temporal evolution of an application landscape which is managed through projects can be analyzed by creating multiple application landscape diagrams with each of them representing one point in time.

3.2.4 Synchronization Management

Again, the PowerDesigner Enterprise Architect module does not support the synchronization of projects, since it does not have a default type for projects. However, synchronization management can be performed by PowerDesigner's integrated Impact and Lineage Analysis functionality, which is able to show the impact of removing or changing an object onto other objects. Thereby, PowerDesigner supports three different representation styles, namely a drop down menu, a table (both Fig. 3.20) and a diagram representation (Fig. 3.21).

3.2.5 Strategies and Goals Management

Since PowerDesigner has neither a predefined strategy nor a predefined goal objects type, it does not support any process of long-term strategy planning.

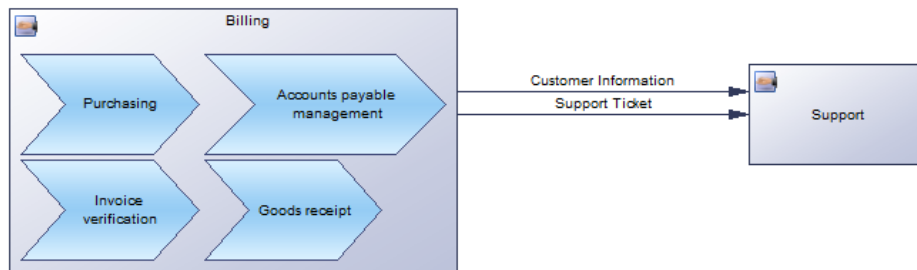


Figure 3.22: PowerDesigner - Business flows

3.2.6 Business Object Management

Although PowerDesigner does not have a predefined object type for business objects, the movement of business objects can be illustrated by using the business communication diagram (c.f. Section 3.1.2). Thereby, this diagram type is used to model business flows between business functions (Fig. 3.22), whereas relations can be annotated to determine the transferred business objects. However, since defining the transferred business objects as annotation, these relations wouldn't have certain semantics. Moreover, while processes and actors can be integrated as well, the integration of applications into this diagram type is not supported.

3.2.7 SOA Transformation

Since providing the SOA diagram type (Fig. 3.9), PowerDesigner supports SOA transformation management. Thereby, business, service, application and technology layers can be modeled. However, like every other diagram type, the SOA diagram cannot be generated automatically, but needs to be created manually.

3.2.8 IT Architecture Management

PowerDesigner's Technology Infrastructure diagram (Fig. 3.26) can be used to perform IT architecture management, especially to model hardware blueprints (Fig. 3.23). However, multiple blueprints need to be modeled in multiple technology infrastructure diagrams.

Due to PowerDesigner's focus on modeling, it lacks some essential features related to IT architecture management. For example, querying the current repository to find out which hardware parts are currently used is not supported.

Nevertheless, the Impact and Lineage analysis (c.f. Section 3.2.4) can be used to show which hardware parts are related to which blueprint (Fig. 3.24). Moreover, impacts caused by architecture changes can be visualized by an impact analysis diagram (Fig. 3.25).

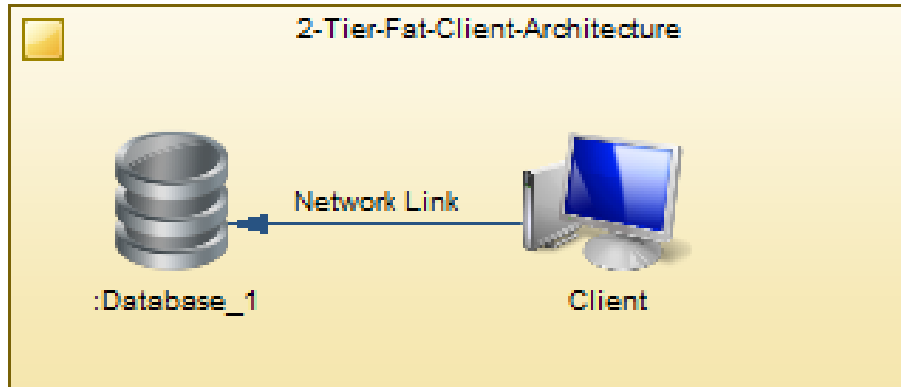


Figure 3.23: PowerDesigner - Blueprint

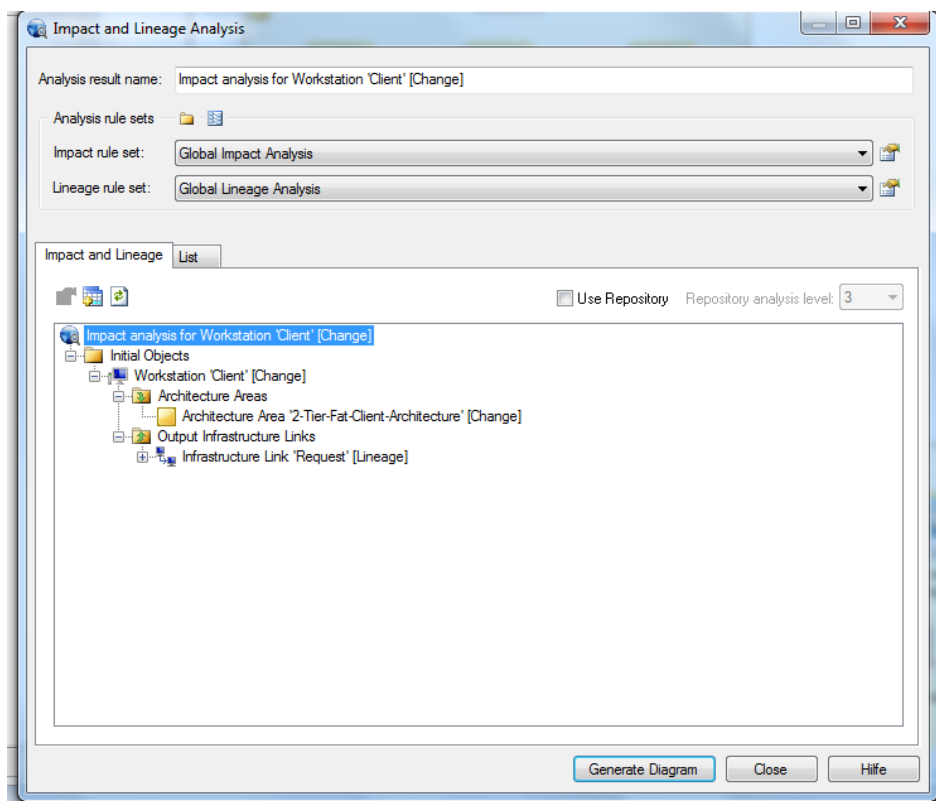


Figure 3.24: PowerDesigner - Related blueprint

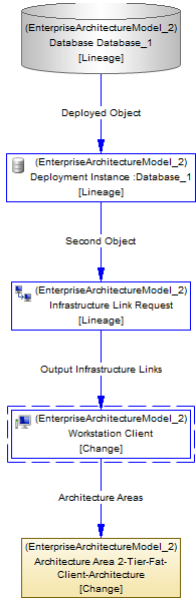


Figure 3.25: PowerDesigner - Impact caused by changed architecture

3.2.9 Infrastructure Management

The PowerDesigner Enterprise Architect module already defines many different hardware object types by default. By using the Technology Infrastructure diagram, PowerDesigner is able to illustrate relationships between applications, hardware and networks (Fig. 3.26). Moreover, additional hardware attributes, e.g. operating costs of a database instance, can be created to extend the existing objects types.

However, infrastructure queries like "What are the current operating costs?" are not supported.

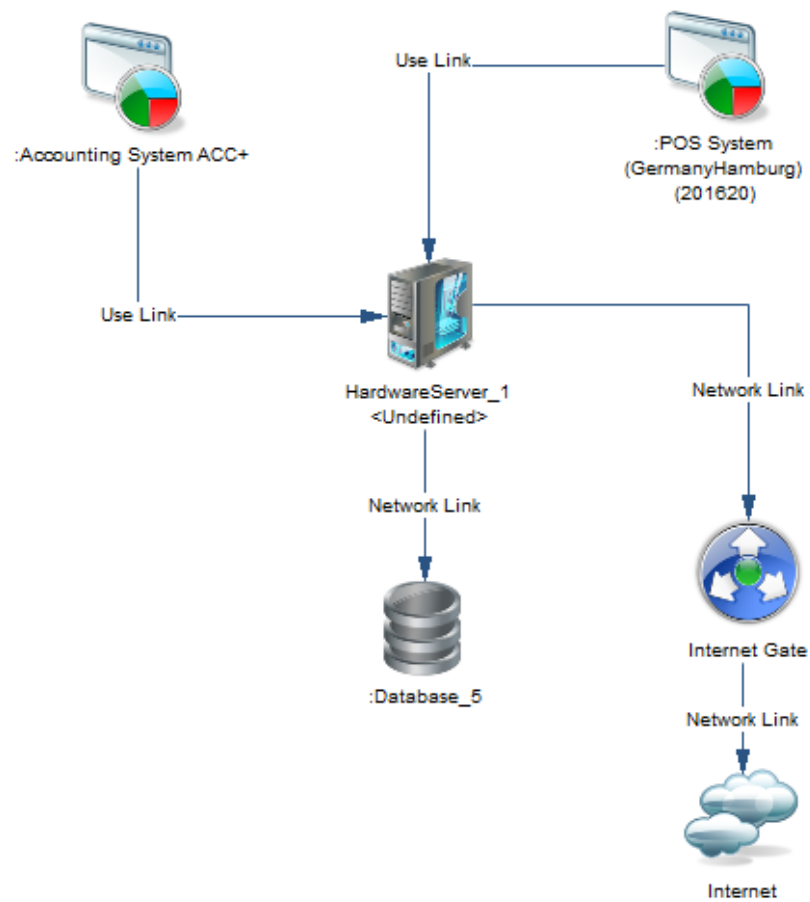


Figure 3.26: PowerDesigner - Infrastructure diagram

Iteraplan is an EAM tool developed by the German company iteratec. It is available in two different versions: community and enterprise edition. The community edition is available as open-source software but only offers a limited set of features. Important features, e. g. data import, are not available in this limited version¹. The enterprise edition is usually sold with a maintenance contract and iteratec consulting activities.

In our evaluation we used the online trial demo version 3.2 that is free of charge for 30 days. In this trial demo most of the enterprise features are available. It is hosted by the vendor and can be accessed through the browser.

4.1 Evaluation of Specific Functionality

In this section the results for simulating the scenarios are presented for the tool *iteraplan*.

4.1.1 Importing, Editing and Validating

The data import function in iteraplan supports Microsoft Excel and XMI file formats. To import data from Microsoft Excel, iteraplan provides an .xlsx template. This template pursues iteraplan's information model. Each tab in the Excel file represents one building block or a connection between two building blocks.

In the first step the corresponding data needs to be entered into the template. Afterwards the saved Excel file can be uploaded into iteraplan and data is being imported. The structure of the uploaded Excel file has to follow iteraplan's template and must not be changed. Fig. 4.1

¹A comparison between community and enterprise edition can be found here: <http://www.iteraplan.de/en/iteraplan-enterprise-edition>, last accessed on: December 1, 2014

4. Iteraplan

shows a successfully performed data import. During the import process data is checked for plausibility, information model changes are checked and inserted data is validated. Information model changes in this situation refer to changed or added building block attributes. During validation phase data is checked e. g. for duplicates. Only after all steps were performed successfully, data is written into iteraplan's database. Thus, either all or no data is imported. Iteraplan does not generate any report that shows which data or objects were imported. Editing data during the import process is not possible.

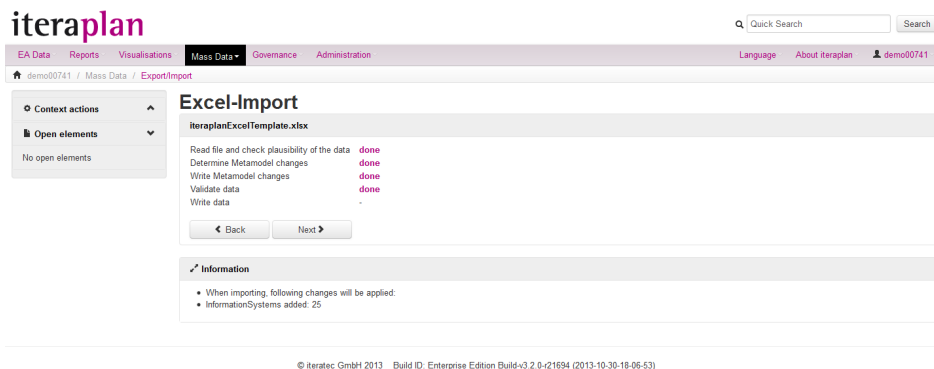


Figure 4.1: Iteraplan - Successful data import

If an error occurs during the import process, iteraplan points out the Excel cell that causes the error (see Fig. 4.2).

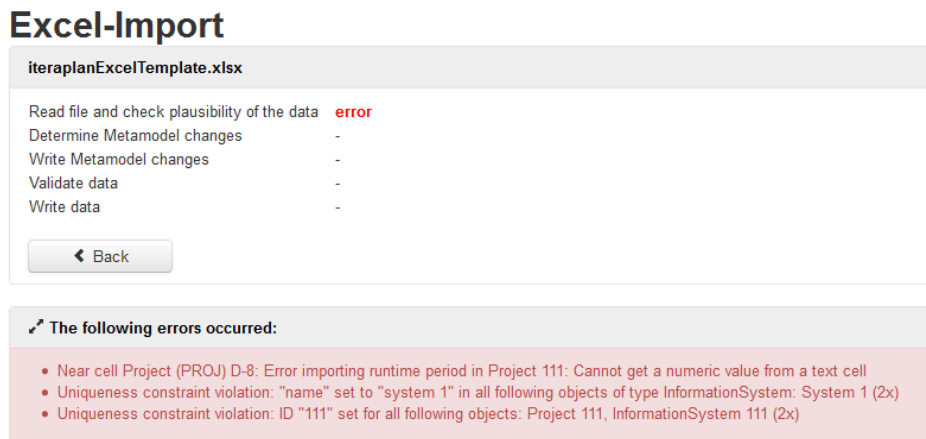


Figure 4.2: Iteraplan - Import error

iteraplan offers the possibility to edit individual info objects manually or a group of objects through the bulk update feature. With the bulk update feature attributes of multiple instances can be updated in one step. This can be useful to change or set the same content for one attribute in multiple information systems. Moreover, data can be exported as Excel and XMI file format. Thus, other tools like Microsoft Excel can be used to edit data. After editing data with external software, data import has to be repeated. Files or URLs that contain relevant

information can be linked to information object instances. Since version 3.2 iteraplan provides a REST interface to extract and read data.

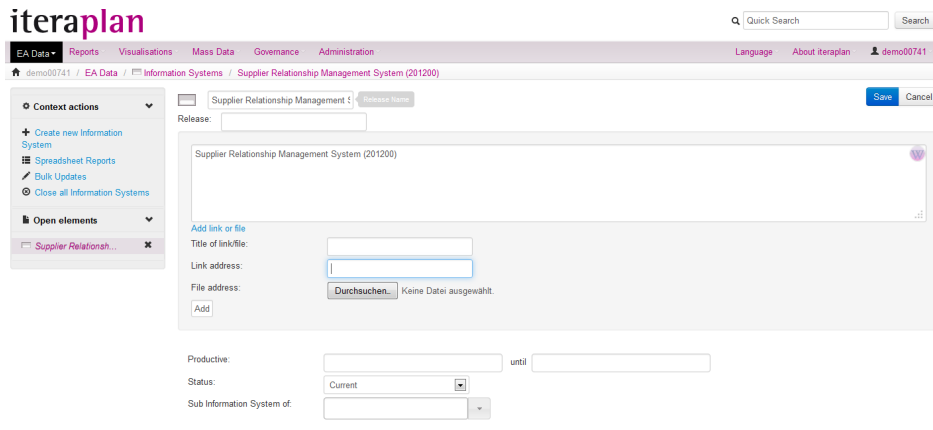


Figure 4.3: Iteraplan - Edit data

4.1.2 Creating visualizations

Iteraplan offers the following visualizations:

- Landscape Diagram,
- Cluster Diagram,
- Nesting Cluster Diagram,
- Portfolio Diagram,
- Master Plan Diagram,
- Bar Chart,
- Pie Chart and
- customizable Dashboard.

In order to generate visualizations it is at first necessary to select the represented information objects. This is done through queries. In these queries filters and selections can be set. As a second step the illustration style can be adjusted. In most diagrams object colors and connection styles can be changed based on attributes. After these settings were made, a diagram is generated.

Iteraplan visualizations can be generated as vector files (svg), images (jpeg, png), document files (pdf) and partly as visio diagrams. The nesting cluster diagram offers two dimensions to select from. Fig. 4.4 represents a nesting cluster diagram that allocates information systems to business units.

In iteraplan's cluster diagram it is possible to select multiple dimensions. Firstly, one dimension needs to be selected and afterwards additional dimensions can be added. Fig. 4.5

4. Iteraplan



Figure 4.4: Iteraplan - Nesting Cluster Diagram

demonstrates which projects, business units and business processes belong to which accounting system. Although not shown in this example, it would be possible to change the color of the objects in the diagram based on an attribute value.

		Information System				
		Accounting and Costing System [1]	Accounting System (500)	Accounting System (S10)	Accounting System (S20)	Accounting System ACC+
Business Process	Project	Connection of Costing and Ac [2]		VAT change in Accounting System	Connection of Costing and Ac [2]	
	Business Unit	VAT change in Costing System			VAT change in Accounting System	
		Headquarter		Headquarter	Headquarter	
		Support Process		Support Process	Support Process	

Figure 4.5: Iteraplan - Cluster Diagram

The portfolio diagram is a powerful diagram where both axis, bubble color and bubble size can be adjusted based on existing attributes. For instance a portfolio matrix as shown in Fig 4.6 can be generated which reflects project urgency, return on investment, costs and pursued demands.

Through the dashboard (Fig. 4.7) animated pie charts can be generated and represented attributes changed on-the-fly. Other available diagram forms are landscape diagrams (Fig. 4.8), information flow diagrams (Fig. 4.9) or master plan diagrams (Fig. 4.10). The individual created bar or pie diagrams can be merged into a composite chart. Swim lane diagrams are not available.

Through the described querying approach many different objects can be plotted against diverse attributes which makes the diagram generation flexible. After generation though, the diagrams are very static and can barely be adjusted.

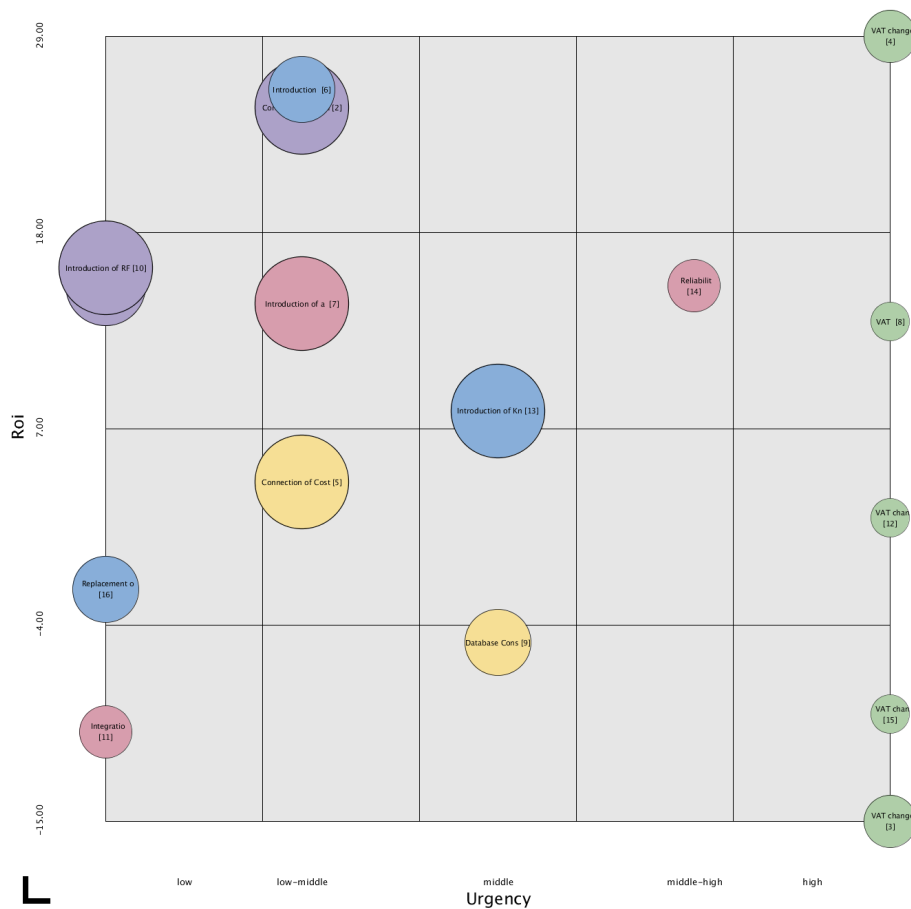


Figure 4.6: Iteraplan - Portfolio Diagram

4.1.3 Interacting with, Editing of, and Annotating Visualizations

During the diagram generation process visualizations can be customized. Color, form and connection between information objects can be adjusted. Fig. 4.11 shows different highlighting options. Customized queries can be saved. Thus, old diagrams can be regenerated quickly.

Changing or adopting generated diagrams is only possible for visio diagrams. For all other read-only output options later editing of the diagrams is not directly possible. The only solution is to generate a new diagram with different settings. Therefore, interaction with the underlying data repository is limited. Semantic changes through external applications are not incorporated. Zoom in/out features and annotations can be performed with external picture editing applications but cannot be integrated back into iteraplan.

4.1.4 Communication and Collaboration Support

Iteraplan includes an user management that supports role-based access control. Different roles and permissions for individual users can be set. Multiple users can be grouped into user groups

4. Iteraplan

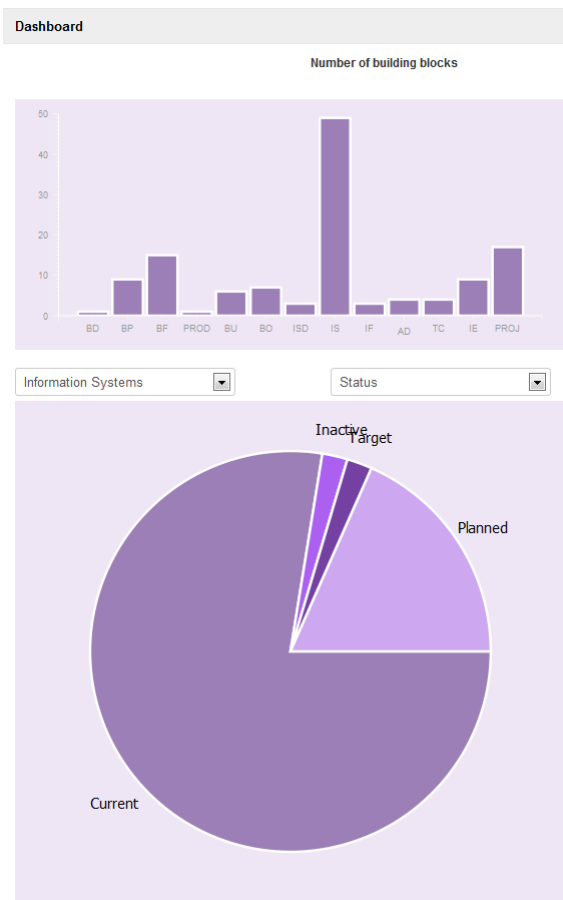


Figure 4.7: Iteraplan - Dashboard

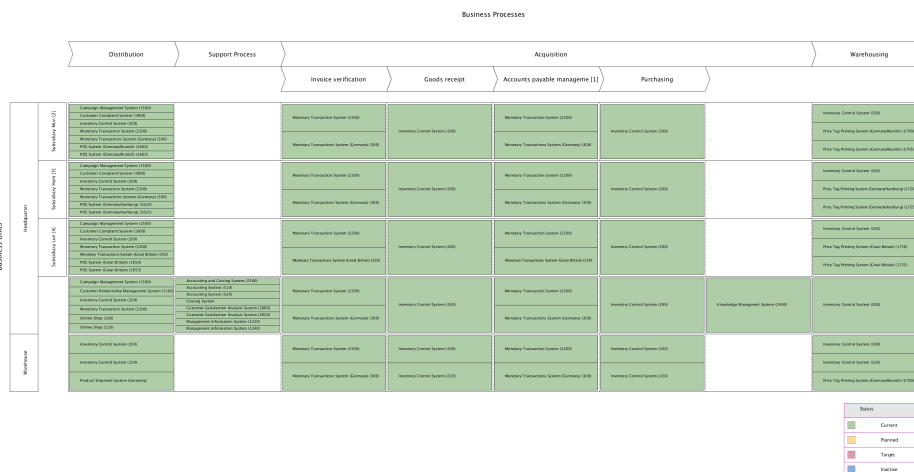


Figure 4.8: Iteraplan - Landscape Diagram

and assigned one or more roles. Moreover, write permissions for specific information instances can be restricted to users or user groups.

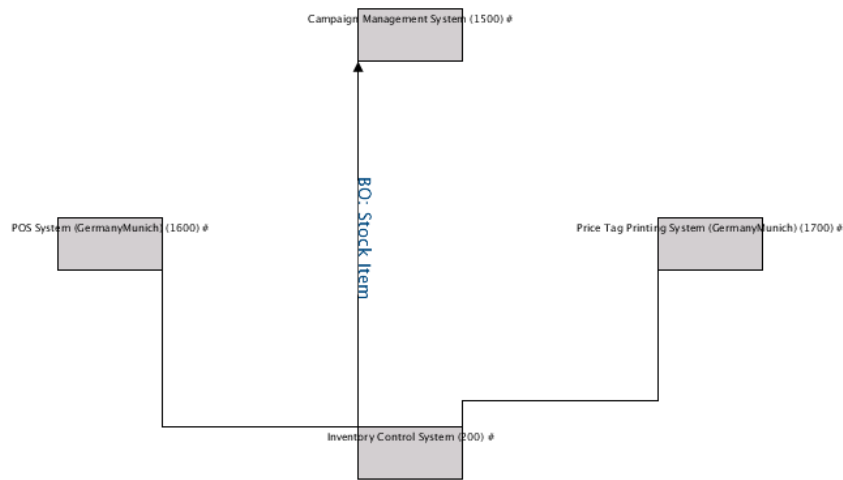


Figure 4.9: Iteraplan - Information Flow Diagram

Information System	Status	Begin	End	2012												2013											
				Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec			
Accounting and Costing Syst	Current	-	-	Productive from / to																							
Accounting and Costing Syst	Inactive	-	02/01/2013	Productive from / to																							
Accounting and Costing Syst	Current	02/01/2013	10/01/2013													Productive from / to											
Accounting and Costing Syst	Planned	10/01/2013	-																								

Information System - Status	
	Current
	Planned
	Target
	Inactive

Figure 4.10: Iteraplan - Master Plan Diagram

The responsibility attribute is a special attribute type. It assigns one user or a user group the responsibility for a particular information object, e. g. a business process or an information system.

In iteraplan every user works through an individual user account and browser session. Collaboration features like chat tools or wikis are not included. The user management functionality inside iteraplan could not be fully tested because the evaluated online trial demo involves only one user.

4.1.5 Flexibility of the Information Model

Iteraplan brings an own best-practice information model that is based on the experiences from customer projects. The model which is explained in books is not yet reproduced completely

4. Iteraplan

Colour settings

Please choose which attributes define the size and colour of the Projects.

- For the dimension **Size**, only numeric attributes and single-value responsibility and enumeration attributes are available.
- For the dimension **Colour**, only numeric, responsibility and enumeration attributes are available.

Size:

Colour:

Adapt to government regulations

Homogenization

Improve Customer retention

Improve enterprise-wide knowledge management

Reduce costs

unspecified

Axis settings

Please choose which attributes shall be mapped to the axes.

X-axis:

Y-axis:

Advanced settings

Output format:

Figure 4.11: Iteraplan - Diagram Customization

Search Users

First name:

Last name:

2 Users found 20 per page

Login name	First name	Last name	E-mail
system	-	-	system@iteraplan.de
test2	Test	User	

Figure 4.12: Iteraplan - User List

in iteraplan. The vendor works on incrementally extending it. The meta model inside iteraplan cannot be changed. New information object types or building blocks cannot be added, although existing building blocks can be renamed.

The only possibility to adjust the existing meta model is to add attributes. Attributes can be added to particular building blocks manually inside iteraplan. Another option is to add attributes through an Excel template and import the new attributes into iteraplan. This option was not available in the evaluated online trial demo and could not be tested. Attributes can be set as mandatory or optional. Multiple values can be included into one attribute. It is possible to edit attributes after creation.

There are different attribute types that can be created: enumeration, numeric, date, text and the earlier mentioned responsibility type. While creating enumerated attributes a default

IS Responsible

Responsible person for IS

[Add link or file](#)

Attribute Type: Responsibility Attribute

Attribute Group: [Default Attribute Group]

Mandatory Attribute:

Multiple Values:

Possible Attribute Values

+ - User -

activated for the following Building Block Types

+ Information System

Figure 4.13: Iteraplan - Responsible Attribute Type

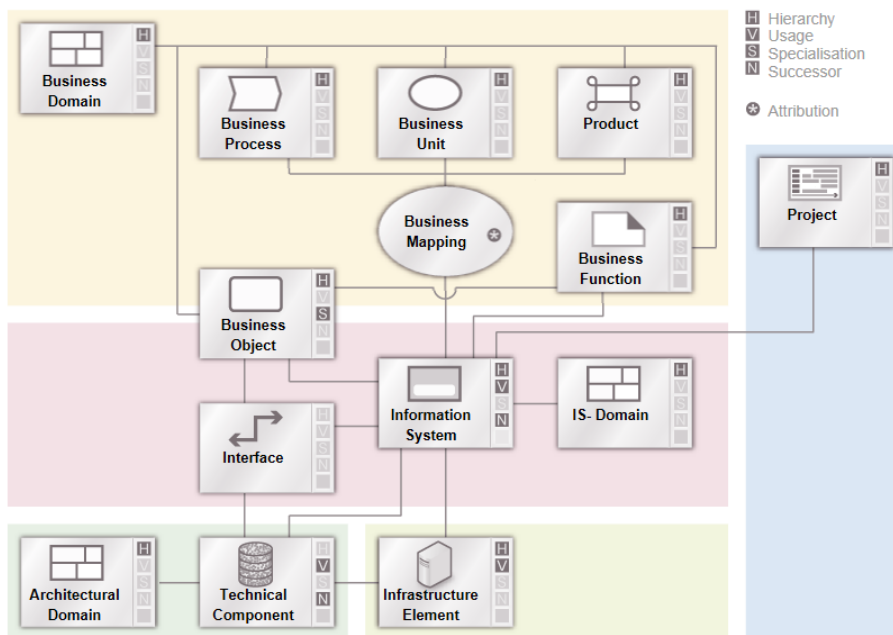


Figure 4.14: Iteraplan - Information Model

color can be set that will be used for graphical representations. A lower and upper bound as well as a unit can be defined for numeric attributes. Attribute instance values can be set for individual information objects or as a bulk process with the bulk update functionality. Another way to edit attributes is to export the data repository into Excel, edit the attributes values with Excel and re-import them into iteraplan. New attributes can be used as query or filter option in diagrams after instance values have been set.

4. Iteraplan

Urgency

Urgency
[Add link or file](#)

Attribute Type: Enumeration Attribute

Attribute Group: ProjectAttributes

Mandatory Attribute:

Multiple Values:

Possible Attribute Values		
Name	Default color	Description
low	Green	low
low-middle	Yellow	low-middle
middle	Blue	middle

Figure 4.15: Iteraplan - Edit Attribute

In conclusion, iteraplan's information is not very flexible and the only possibility to adjust the model to individual requirements is to add attributes. Attributes cannot reflect all changes. Therefore, the missing possibility of adding information object types makes it difficult to adapt an own information model into iteraplan.

4.1.6 Support of Large Scale Data

Iteraplan provides good support for large scale data. Large amounts of objects can be imported through the MS Excel import interface. The system works smoothly and its response time remains short.

The procedure to create visualizations remains the same. First, the data to be represented are selected through queries and then the visualization with all selected objects and predefined settings is created automatically. Single objects can be found and accessed quickly through the search functionality. The bulk update functionality is especially useful to update attributes of large amounts of objects easily.

4.1.7 Reporting

Iteraplan offers the possibility to generate spreadsheet reports. The output format differs from the building block type that is looked at. Besides a simple table potential output file formats include Microsoft Excel, XML, CSV, Microsoft Project XML Format and Gantt Project MPX Format. Report generation is similar to diagram generation: information objects can be selected and filtered out depending on attribute values or search queries.

Results: 16							+ Add Column
Name and Version	from	until	Cost	pursuesDemand	Roi	Urgency	
⌂ ▶	◀ ⌂ ▶	◀ ⌂ ▶	◀ ⌂ ▶	◀ ⌂ ▶	◀ ⌂ ▶	◀ ⌂ ▶	
VAT change in Price Tag Printing System	06/15/2008	07/01/2008	40.00	Adapt to government regulations	13.00	high	
VAT change in POS System	04/01/2008	04/15/2008	80.00	Adapt to government regulations	-15.00	high	
VAT change in Costing System	01/10/2008	05/15/2008	50.00	Adapt to government regulations	2.00	high	
VAT change in Accounting System	10/01/2008	11/01/2008	100.00	Adapt to government regulations	29.00	high	
Introduction of RFID in the Warehouse	02/01/2008	08/14/2008	1500.00	Reduce costs	16.00	low	
VAT change in Online Shop	02/01/2008	10/01/2008	60.00	Adapt to government regulations	-9.00	high	
Reliability Improvement of the Online Shop	02/01/2008	11/01/2008	70.00	Improve Customer retention	15.00	middle-high	
Introduction of Knowledge Management	02/15/2008	08/15/2008	650.00	Improve enterprise-wide knowledge management	8.00	middle	
Database Consolidation	04/05/2008	09/15/2008	150.00	Homogenization	-5.00	middle	
Consolidation of Monetary Transaction Systems	04/15/2008	06/15/2008	2000.00	Reduce costs	25.00	low-middle	
Introduction of a Bonus Card for Customers	04/01/2008	10/01/2008	750.00	Improve Customer retention	14.00	low-middle	
Introduction of a new Management Information System	02/01/2009	06/15/2009	400.00	Improve enterprise-wide knowledge management	26.00	low-middle	
Replacement of the Customer Satisfaction Analysis System by the Data...	03/01/2007	09/30/2007	190.00	Improve enterprise-wide knowledge management	-2.00	low	
Optimization of Monetary Transaction System	04/15/2008	06/15/2008	600.00	Reduce costs	15.00	low	
Connection of Costing and Accounting System	01/15/2008	10/15/2008	1000.00	Homogenization	4.00	low-middle	
Integration of an auctioning platform into the online shop	02/01/2008	11/30/2008	100.00	Improve Customer retention	-10.00	low	

Figure 4.16: Iteraplan - Generated Report

Moreover, iteraplan includes an own querying language called iteraQL that can be used for filtering or abstracting specific information. This language allows joins, compare operations, boolean operators and some property operators like count. The syntax shows similarities with XPath.

Another reporting tool is the portfolio diagram (Fig. 4.6) that can be used to highlight information objects like projects by bubble color and bubble size. The information flow diagram (Fig. 4.9) illustrates how information system are interconnected. It can be used to determine one information system's influence on others. The before mentioned REST interface could also be used to extract data from iteraplan and generate reports with external tools.

Predefined KPIs about the application landscape or other EA critical figures are not available. One complicated way to generate KPIs could be to use iteraQL or REST for data extraction and then define own KPIs in external tools like Excel.

4.1.8 Usability

Iteraplan is always accessed through a web browser and can therefore be used like a usual web page. The tool offers a clear structure with a good overview. During our evaluation system response times were usually quick and no big delays were noticed. Iteraplan offers a useful documentation in form of a wiki² that eases the use and is a helpful resource if problems occur.

The possibility to save queries and re-generate diagrams is helpful in daily use and allows to gather up-to-date EA information quickly. Setting up proper diagrams in the beginning takes some time, though. The approach to first select the represented data and then adjust represented attributes and graphical style is structured but time consuming. Iteraplan does not offer any predefined, fully automatically generated diagrams. Another shortcoming is the missing interaction between diagrams and underlying data repository. The bulk update functionality saves time when multiple instances shall be assigned the same attribute value.

²<http://www.iteraplan.de/wiki/display/iteraplan/iteraplan+Documentation+Home>, last accessed on: December 1, 2014

4.2 Evaluation of EA Management Support

In this section iteraplan’s EA Management support capabilities are evaluated in detail.

4.2.1 Landscape Management

Information systems in iteraplan come with the attribute 'status'. Possible values for this attribute field are 'current', 'planned', 'target' and 'inactive'. Another attribute to support landscape management is the version number that every information can get assigned. Iteraplan offers the possibility to indicate application hierarchies: parent, predecessor and successor systems can be set. Information systems can be mapped to business architecture, technical architecture and projects. This functionality allows to look up which information system is used in which business process. Moreover, the time period a system will be productive can be set.

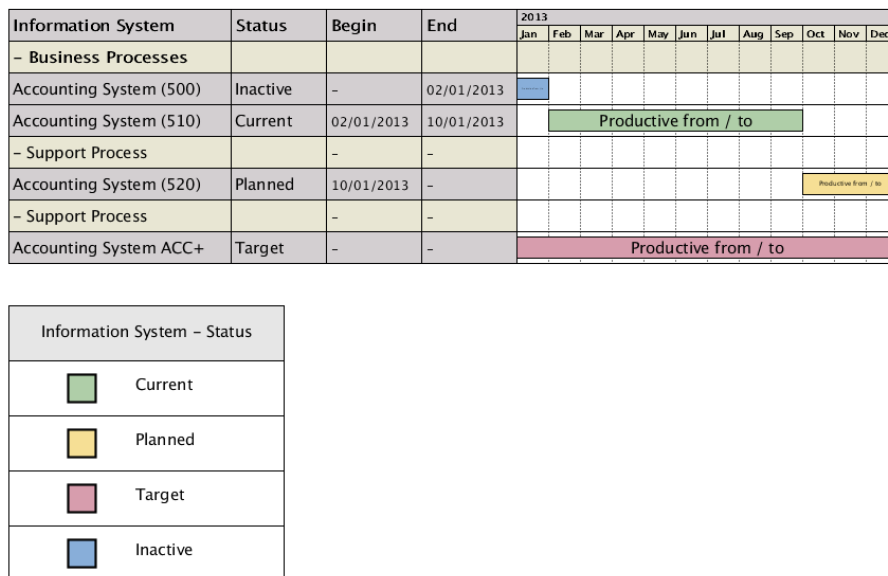


Figure 4.17: Iteraplan - Master Plan Diagram representing Change over Time

The landscape diagram shown in Fig. 4.8 can be used to illustrate the application landscape in connection with other building blocks, e. g. business processes or projects. Different statuses can be displayed but time periods can not be selected. Thus, the landscape diagram only represents the current landscape. Change over time can be represented with the master plan diagram. The illustrated time span needs to be set before the diagram is generated. Statuses can be highlighted through different colors. Fig 4.17 shows an example diagram where different accounting systems are used over time.

It is complicated in iteraplan to compare a current landscape with a planned landscape. Two diagrams need to be generated, one with current and one with planned information systems. Multiple planning scenarios can only be illustrated by adding more information systems of the type but different version numbers or creating manually a new attribute 'planning variant'.

4.2.2 Demand Management

Demand management is not directly available as a dedicated function in iteraplan. Demands can be integrated by adding attributes like 'pursuesDemand' to projects. In this way, reports and diagrams can be manually generated that illustrate which projects pursue which demands.

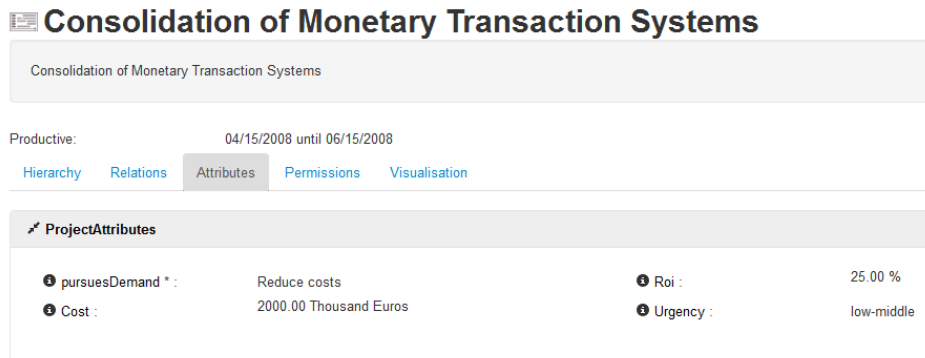


Figure 4.18: Iteraplan - Project Attribute 'pursues Demand'

Demand building blocks are not available. Thus, connections between demands and information systems are not existent. Affected information systems can be discovered by looking up project details. Demands cannot be linked to goals or strategies because they are not modeled in iteraplan. A demand repository can not be set up. The only possibility to integrate new demands is to change or extent manually created attributes.

4.2.3 Project Portfolio Management

Iteraplan's meta model includes a particular object type for projects. Projects can be created inside iteraplan with the attributes description, time, affected information systems and subordinate projects. Additional attributes like costs, ROI or urgency need to be added manually. These data can be used to generate reports and diagrams.

An example report that lists all existing projects and with selected attributes can be found in Fig. 4.16. Besides a simple table other output formats like Microsoft Project XML format are available. Filters by time, urgency or any other attribute can be applied before generating a report. Projects can be represented in a project portfolio diagram where attributes can be highlighted by x- and y-axis, bubble color or bubble size (Fig. 4.6). The nesting diagram can be used to show which information systems, business processes or business units are affected by which projects.

4.2.4 Synchronization Management

Regarding synchronization management iteraplan offers the possibility to model project dependencies through parent/child relationships. Nested cluster diagrams can illustrate organizational units or information systems that are affected by projects. Master plan diagrams

4. Iteraplan

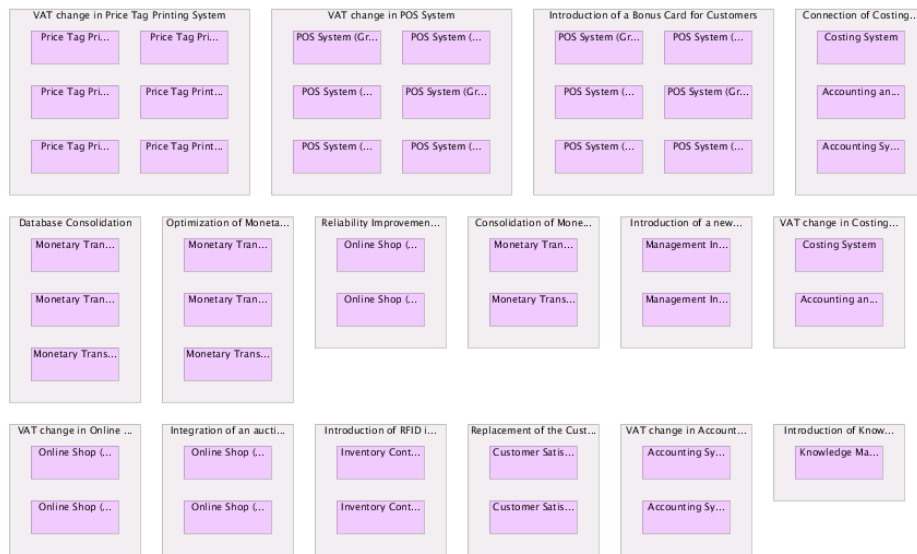


Figure 4.19: Iteraplan - Affected Information Systems by Project

can represent different projects over time. Project delays can be included into iteraplan by extending the project time span. The appearance of the project does not change, though. It is not highlighted as 'delayed'. Delays or conflicting projects are not well illustrated in iteraplan. Interaction with elements is not possible in read-only generated diagrams.

4.2.5 Strategies and Goals Management

Organizational strategies and goals are not integrated in iteraplan's meta model. New object types representing this organizational area cannot be defined. The only way to integrate strategies is to add attributes manually to object types, similar as for demand management. After adding attributes like 'strategic value' or 'strategy driver' to information systems or projects, these new attributes could be used in portfolio diagrams. A complete drill-down from strategy to goals to information system is not possible with this approach, though.

4.2.6 Business Object Management

Business objects are integrated in iteraplan's meta model. Operations performed on business objects are modeled through business functions. Multiple business objects can be assigned to one business domain. The information flow diagram (Fig. 4.9) displays which business objects are transferred between which information system by which technical interface.

4.2.7 SOA Transformation

The service concept is not integrated in iteraplan. Services could be modeled as planned information systems that will replace other IS in the future. SLAs like 'minimum availability'

could be modeled as attributes. Transformation projects that replace old information systems with new services can be created.

It is difficult in iteraplan to discover potential information systems where service replacements would be useful. The count operator from iteraQL could be used to get the amount of business processes that use one particular information system. Predefined reports or diagrams that show this number are not available. Applying a top-down approach, nesting cluster diagrams (Fig. 4.4) are helpful to determine information systems that are used by many business entities. The identified systems could be replaced by new services. The master plan diagram (Fig. 4.10) could represent current information systems and future services.

4.2.8 IT Architecture Management

Iteraplan provides the meta model building block 'Architectural Domain' for IT architecture management. Architectural domains represent standardized architectural blueprints. These blueprints can be modeled through technical components that can be added into iteraplan. Productive information systems are based on technical components and run on infrastructure elements. Architectural attributes like 'availability' need to be added manually.

✦ Technical Architecture	
Is based on the following Technical Components	
Name	Description
ApplicationServer	ApplicationServer
Database	Database

✦ Infrastructure Architecture	
Runs on the following Infrastructure Elements	
Name	Description
Apache 2.0.53	Apache 2.0.53
Oracle 9i	Oracle 9i
Tomcat 5.1	Tomcat 5.1

Figure 4.20: Iteraplan - Information System's underlying technologies

Cluster diagrams (Fig. 4.21) can illustrate how architectural blueprints make use of technical components.

(AND)

Properties of assigned Infrastructure Elements

No Associations

Name contains Apache 2.0.53

(AND) OR

<Add query extension>

Rework query results

Output: Simple list

Send Query Reset

Save query...

Results: 1 + Add Column

Name and Version	Description	from	until	Status
✖ ▶ Campaign Management System (1500)	◀ ✖ ▶ Campaign Management System (1500)	◀ ✖ ▶	◀ ✖ ▶	◀ ✖ ▶ Current

Figure 4.23: Iteraplan - Infrastructure Elements Spreadsheet Report

Infrastructure Element

IE 6.0	Apache 2.0.53	MySQL 2.1	DB2 6.0	Toncat 5.1	Oracle 9i	Proprietary Fat-Client	Rea Weblogic 6.1
Accounting and Costing System (12)	Business Traveling System (2) (1)	Business Traveling System (2) (1)	Accounting and Costing System (2)	Accounting and Costing System (2)	Campaign Management System (1)		
	Campaign Management System (1)			Campaign Management System (1)			

Figure 4.24: Iteraplan - Technical Components per Information System

