



Service Provisioning with Model Guardian

Model Guardian includes a separate add-in called Service Provisioning, which follows and extends the service concept from TOGAF. The idea is that elements in higher architecture layers require services of the elements in lower architectural layers. For example, a claims process in the business architecture requires a batch claims processing service to run each evening and an ad hoc claims processing service for those that need to be done by a claims rep. Likewise, elements from the information systems architecture layer require services of the infrastructure (aka technology) architecture layer. Examples include an execution environment service to run an application and a login service to validate users.

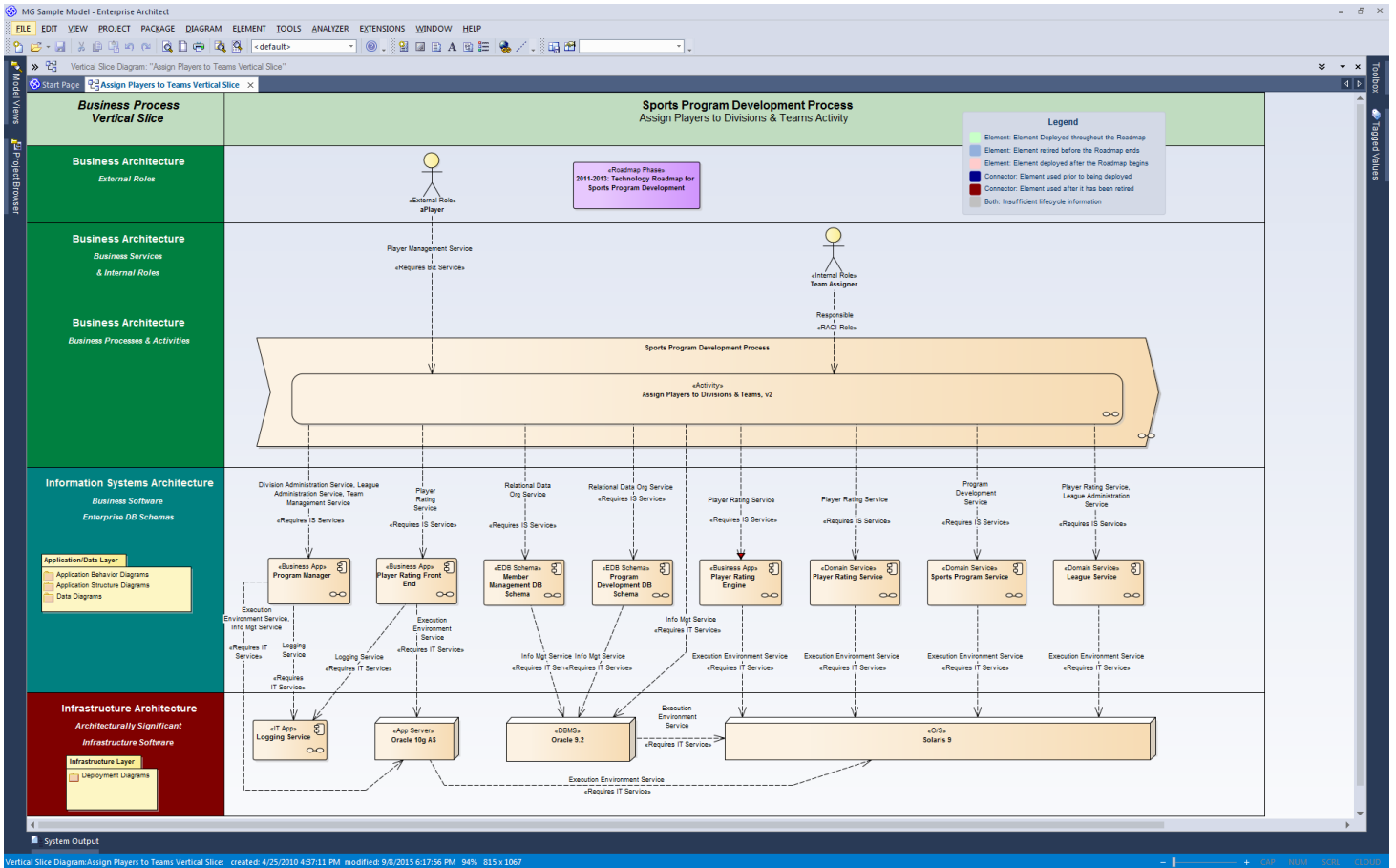
These architectural services are a superset of SOA services. The latter are realized by software, but the former may or may not be. For example, an application may require a programming service, e.g. C#, for its maintenance. This type of service would never become an SOA service (unless we get much more sophisticated than we are now!). The services can be considered architectural requirements. They may also be considered as synonymous with capabilities.

Model Guardian provides an automated approach to maintaining the services and the service elements, e.g. business activities, applications, DBMSs, operating systems, etc. It helps you build a service hierarchy and export it into the EA models. It also lets you assign the services, as tagged values, to the service elements as required services and provided services. Then, it assists in creating service provisions, i.e. relationships between a service client, a service provider, and the connector between the two. When creating the connector via the MG user interface, it will automatically add the specific service(s) required by the client and provided by the provider as a tagged value(s) and also adds the service name(s) to the connector's label.

The following diagram shows a Business Process Realization, aka. Vertical Slice, diagram. This type of diagram shows the architectural elements required to satisfy a particular business activity. The diagram comes from a sample model used to illustrate the features of MG. As you can see, the services in play appear on the connectors.



Service Provisioning with Model Guardian

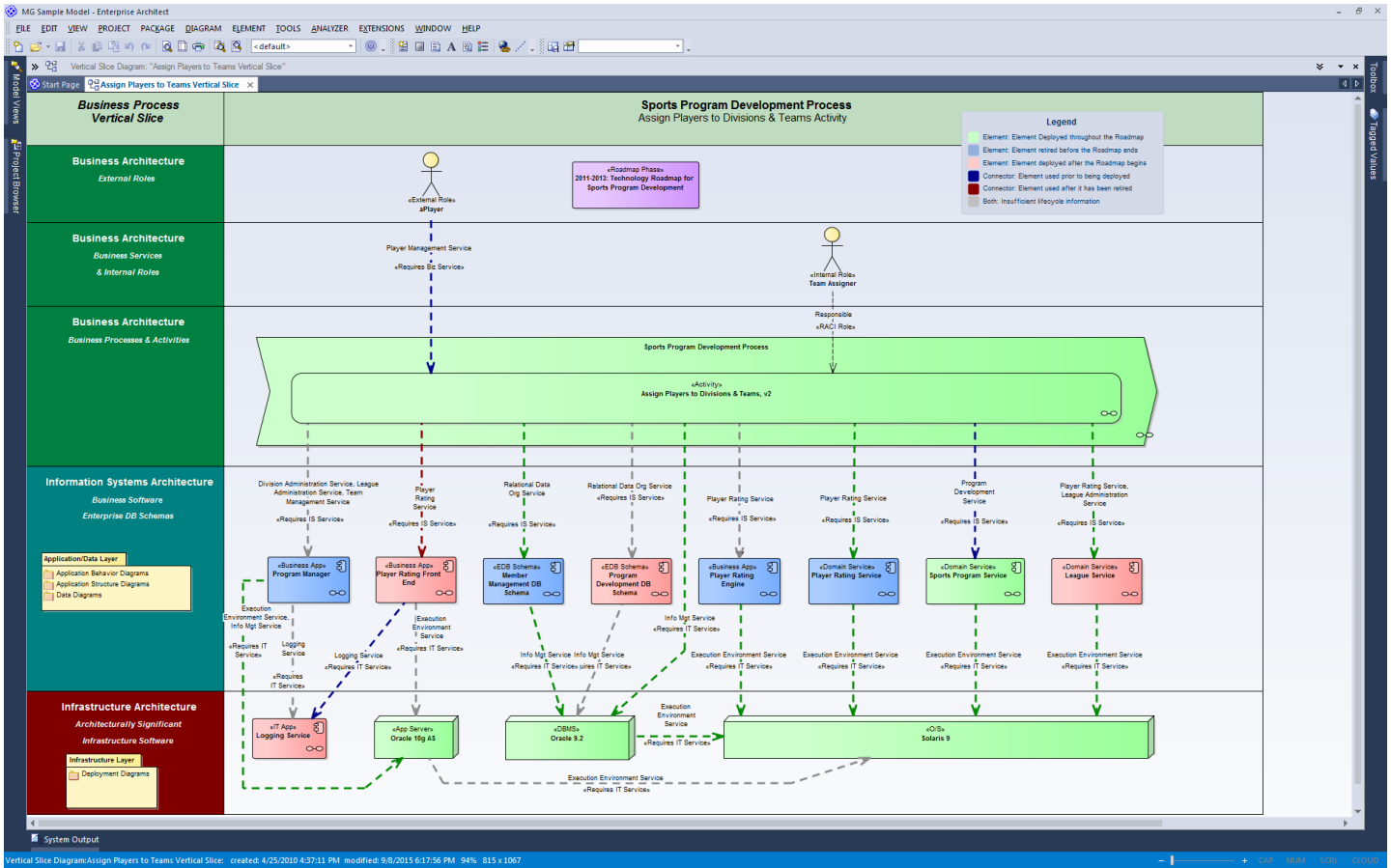


In addition to the services, the service elements and service provisions are given lifecycle dates; namely, Planned Begin Date, Planned End Date, Actual Begin Date, Actual End Date. This lets us know when a particular service element is available for use. By having the lifecycle tags on the service provision connectors as well, we know when the client will/is/was getting the service from the provider.

With this, we have enough information to do some planning. The diagram above has a Roadmap Phase element on it (toward the top in purple). In this context, a roadmap provides the purpose and general information about a plan to move the architecture forward. To implement the roadmap, various Roadmap Phases are set up with begin and end dates. Roadmap Phases allow us to create an incremental plan to implement the overall Roadmap. We can use MG to compare the dates of the Roadmap Phase (or any dates that we provide) to the lifecycle information of the elements and connectors on the diagram. The following is the same diagram but color coded for the dates of the roadmap phase. There is a color legend in the top right of the diagram.



Service Provisioning with Model Guardian



The color map is controlled by the Deployment Map tab of the Service Provisioning dialog shown next.



Service Provisioning with Model Guardian

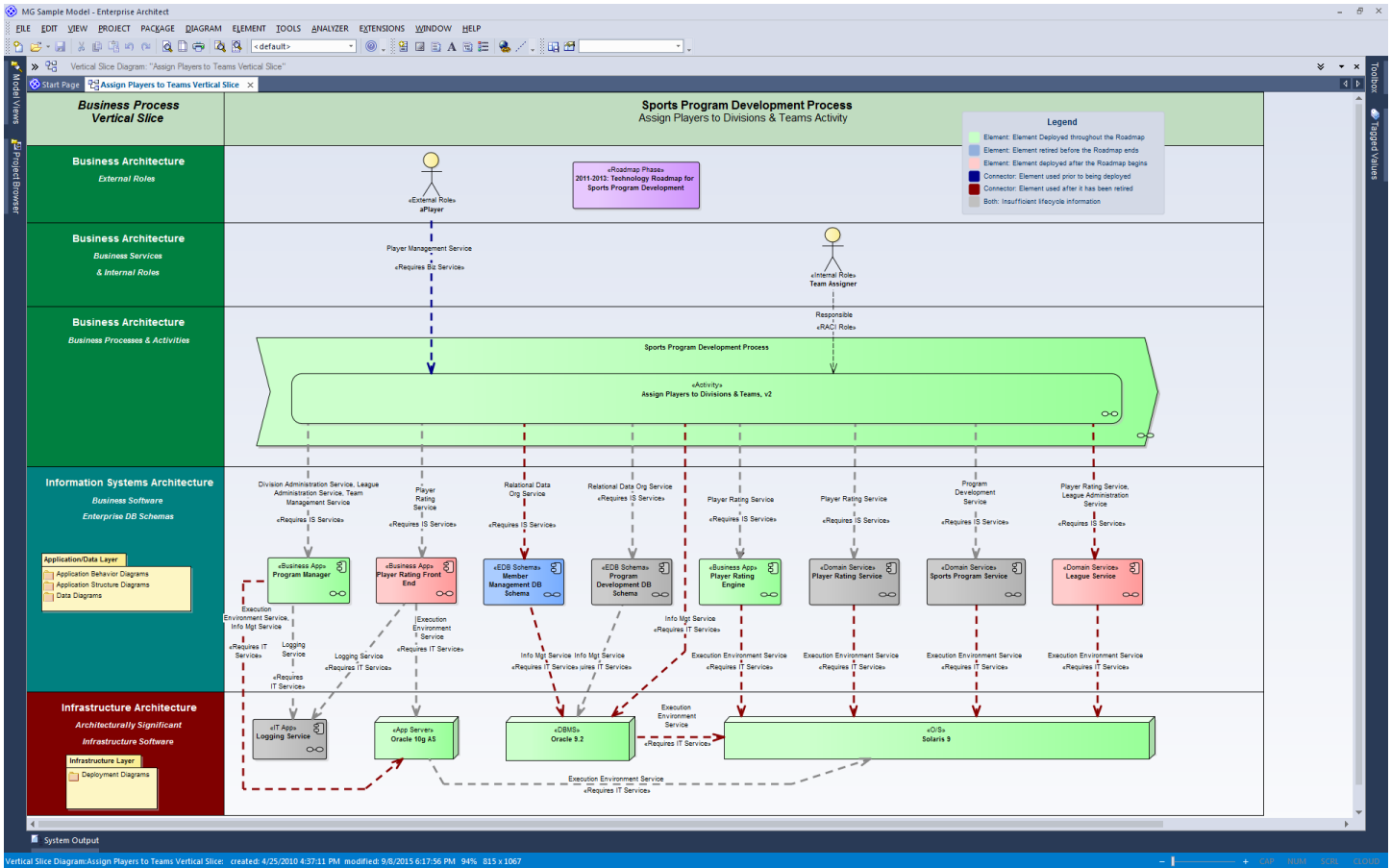
The screenshot shows the 'Service Maintenance' application window. The title bar reads 'Service Maintenance'. The main window has a tabbed interface with 'Service Maintenance' and 'Service Provisioning' tabs. Under 'Service Provisioning', there are sub-tabs for 'Service Provision Management' and 'Deployment Map'. A 'Get from EA' button is in the top right. The 'Diagram Parameters' section includes radio buttons for 'Planned' (selected), 'Current', and 'Normal'. It also has input fields for 'Period Start Date' (2011-01-01) and 'Period End Date' (2013-12-31), along with 'Process' and 'Help' buttons. A note states: 'If the dates are left blank, only the Connectors' lifecycle dates will be verified against the supplying Elements' lifecycle.' Below this is a table with 'Elements' and 'Connectors' tabs. The 'Elements' tab is active, showing a table with columns: Element, Error Condition, Actual Start Date, Actual End Date, Planned Start Date, and Planned End Date. The table contains several rows of data, including 'League Service', 'Logging Service', 'Member Management DB Schema', 'Player Rating Engine', 'Player Rating Front End', 'Player Rating Service', 'Program Development DB Schema', and 'Program Manager'. At the bottom of the window are 'Help', 'Close', and 'Keep on Top' buttons.

| Element | Error Condition | Actual Start Date | Actual End Date | Planned Start Date | Planned End Date |
|-------------------------------|---------------------------------------|-------------------|-----------------|--------------------|------------------|
| League Service | Element deployed after Roadmap begins | 2013-01-01 | | 2014 | 2015 |
| Logging Service | Element deployed after Roadmap begins | | | 2014-03 | |
| Member Management DB Schema | Element retired before Roadmap ends | 1998 | 2000 | | |
| Player Rating Engine | Element retired before Roadmap ends | 2008 | | | 2013-03-01 |
| Player Rating Front End | Element deployed after Roadmap begins | 2012-01-31 | | 2011-01-01 | 2014 |
| Player Rating Service | Element retired before Roadmap ends | | | 2011-01-13 | 2013-07-01 |
| Program Development DB Schema | Element deployed after Roadmap begins | | | 2016 | |
| Program Manager | Element retired before Roadmap ends | 2008-06-01 | | 2008 | 2012-01-01 |

You can see the dates from the roadmap phase at the top. For the color blind and those wanting a little more information, any anomalies are listed in the grid. Elements that do not fully conform to the roadmap phase are listed. The anomalous connectors are shown below.



Service Provisioning with Model Guardian



MG has SQL views that pull this info out of the repository. For example, here is report that shows the deployment status of the configuration items over a 5 year period.



Service Provisioning with Model Guardian

SCMS Reports - Microsoft Visual Studio (Administrator)

File Edit View Project Build Debug Tools Test Window Help

Enterprise Data Flow Diagrams.rdl [Design] Enterprise Data Flow.rdl [Design] Configuration Items.rdl [Design] Roadmapped BP Rea...nts.rdl [Design] Service Driven BP ...ocess.rdl [Design] Five Year Plan.rdl [Design] Project Context.rdl [Design]

Design Preview

1 of 1 100% Find | Next

Enterprise Architectural Service Provider Five Year Plan

| Service Layer | Service Category | Service | Service Specialization | Service Provider | 2012 | 2013 | 2014 | 2015 | 2016 |
|--------------------------|---------------------------------|---|--|--|-------------------------|---------------------------|---------|---------|---------|
| IS Service | Club Management IS Services | Coach Management Service | | Program Manager | | Commissioned | Current | Current | Current |
| | | Member Management Service | | Member Management Application | Current | Decommissioned | | | |
| | Program Development IS Services | Division Administration Service | | Program Manager | | Commissioned | Current | Current | Current |
| | | League Administration Service | | Program Manager | | Commissioned | Current | Current | Current |
| | | Player Management Service | | Program Manager | | Commissioned | Current | Current | Current |
| | | Player Rating Service | | ACME Rating Application Player Rating Engine Player Rating Front End | | Commissioned | Current | Current | Current |
| | | Program Development Service | | Program Manager | | Commissioned | Current | Current | Current |
| | Team Management Service | | Program Manager | | Commissioned | Current | Current | Current | |
| | Scheduling IS Services | Divisional Scheduling Service | | Season Scheduler Application | Current | Current | Current | Current | Current |
| | | Team/Player Statistics Tracking Service | | Stats Manager | Current | Current | Current | Current | Current |
| Venue Management Service | | | Venue Manager | Current | Current | Current | Current | Current | |
| IT Service | Communication Services | Telecommunication Services | FAX Service | Alchemy RightFax | Commissioned Current | Current Decommissioned | Current | Current | Current |
| | Data Management Services | Info Mgt Service | DBMS Service | Oracle 9.2 | Current | Current | Current | Current | Current |
| | Execution Environment Services | Execution Environment Service | DBMS Service | Oracle 9.2 | Current | Current | Current | Current | Current |
| | Process Management Services | Process Control Service | Transaction Process Monitoring Service | CICS | Current | Current | Current | Current | Current |
| | SDLC Services | S/W Programming Service | C# Programming | Visual C# | Current | Current | Current | Current | Current |
| C++ Programming | | | Visual C++ | | | | | | |

Here is a quick tour of the other tabs in the Service Provisioning add-in.

The Service Maintenance tab lets you define the service hierarchy and then export it into the current EA model. Should you make changes in the model, you can sync them back to the service hierarchy in MG. The next image shows the service hierarchy on the left and the editor for the services on the right.



Service Provisioning with Model Guardian

The screenshot displays the 'Service Maintenance' application window. The title bar reads 'Service Maintenance'. Below the title bar, there are two tabs: 'Service Maintenance' and 'Service Provisioning'. A 'Get from EA' button is located in the top right corner.

The main area is divided into two panes. The left pane, titled 'Service Hierarchy', shows a tree view of services. The right pane, titled 'Service Elements', is used for configuring a specific service.

Service Hierarchy (Left Pane):

- Biz Services
- IS Services
- IT Services
 - Auditing Services
 - Accounting Auditing Service
 - Information Processing Facility Audit Service (R/P)
 - System Audit Service (R/P)
 - Authorization Service (R/P)
 - Business Intelligence Services
 - Analytics Service (R/P)
 - Benchmarking Service (R/P)
 - Business Performance Management Service (R/P)
 - Data Mining Service (R/P)
 - OLAP Service (R)
 - Hybrid OLAP Service (P)
 - Multidimensional OLAP Service (P)
 - Relational OLAP Service (P)
 - Predictive Analytics Service x (R/P)
 - Reporting Service
 - Client/Server Reporting Service (R/P)
 - Decision Support Service (R/P)
 - Embedded Application Reporting Service (R/P)
 - Integrated Reporting Service (R/P)
 - Self-Service Reporting Services (R/P)
 - Text Mining Service
 - Communication Services asdsetwe
 - Data Communication Service (R/P)
 - RFID Service (R/P)
 - Telecommunication Services
 - FAX Service (R/P)
 - VOIP Service (R/P)
 - Data Interchange Services
 - Document Generic Data Typing and Conversion Service (R/P)
 - Electronic Data Interchange Service (R/P)
 - ETL (R/P)
 - Graphics Data Interchange Service (R/P)

Service Elements (Right Pane):

Name: Client/Server Reporting Service

Parent: Reporting Service Change Parent Service

Service Dates

| ---- Planned ---- | | ---- Actual ---- | |
|----------------------|----------------------|----------------------|----------------------|
| Begin Date: | End Date: | Begin Date: | End Date: |
| <input type="text"/> | <input type="text"/> | <input type="text"/> | <input type="text"/> |

Notes: Individual installations of reporting tools are placed on a user's machine, in support of personal data retrieval requirements . Source: Cizer Software Corporation.

Use as Required Service Use as Provided Service

Add Child Service Delete Service Cancel Changes Save Service

For a Service to be used as a required Service
- It may have no ancestor marked as Required
- If it has any child Services, they must be marked as Provided

For a Service to be used as a provided Service
- It may not have any child Services
- Either its parent Service or itself must be marked as Required

Buttons: Help, Close, Keep on Top

Next, the Service Elements can be given the services they require and those they provide in the editor on the right (see below).



Service Provisioning with Model Guardian

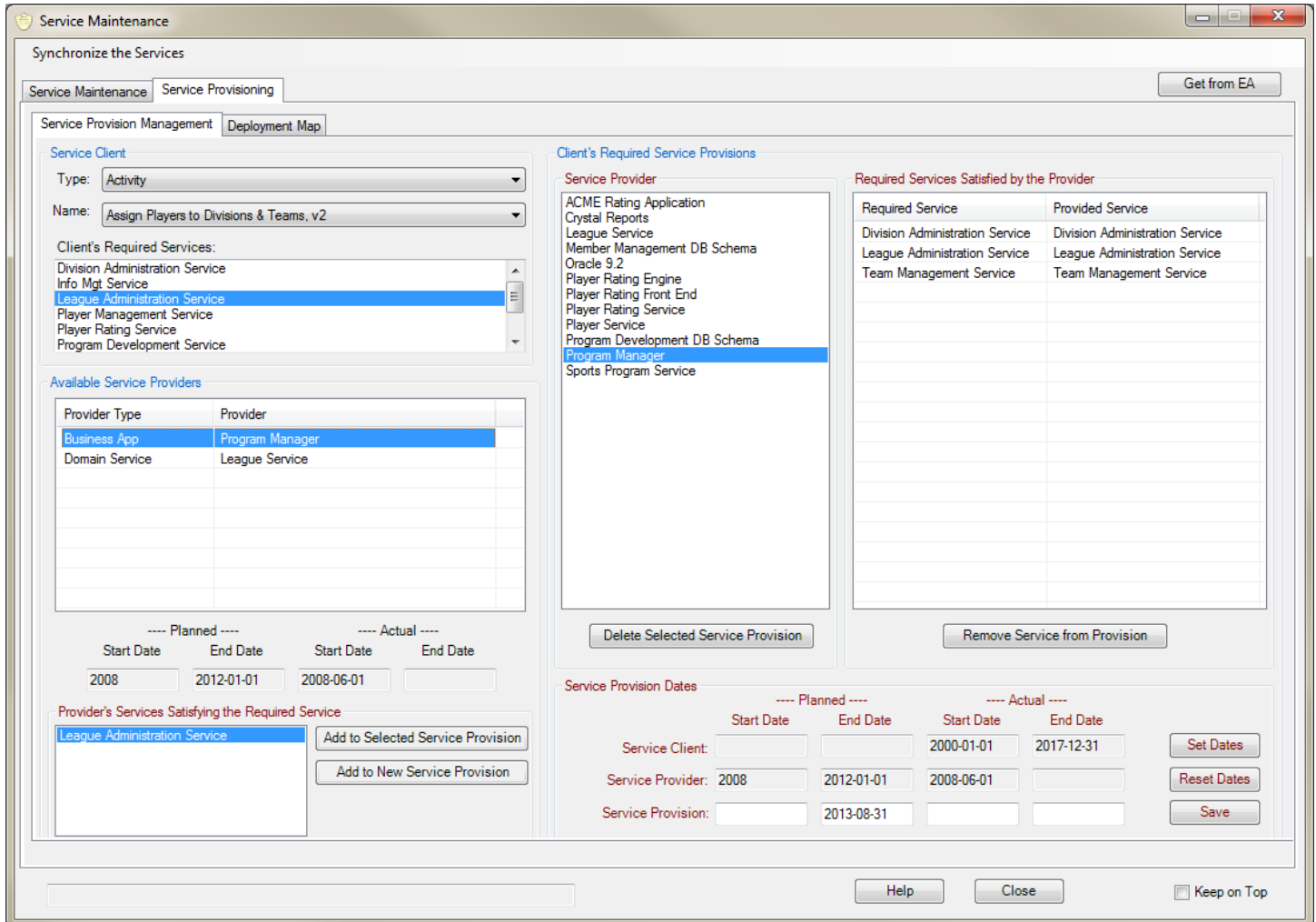
The screenshot displays the 'Service Maintenance' application window, specifically the 'Service Provisioning' tab. The interface is divided into several sections:

- Service Hierarchy:** A tree view on the left showing a list of services under categories like 'Biz Services', 'IS Services', and 'IT Services'. The 'Reporting Service' is currently selected.
- Services / Service Elements:** A central area with two tabs. The 'Service Elements' tab is active, showing:
 - Element Type:** Activity
 - Element Name:** Assign Players to Divisions & Teams, v2
 - Deployment Dates:** A section for 'Planned' and 'Actual' dates. The 'Planned' section has empty fields for 'Begin Date' and 'End Date'. The 'Actual' section has 'Begin Date' set to 2000-01-01 and 'End Date' set to 2017-12-31. There are 'Reset Dates' and 'Save Dates' buttons.
 - Required Services:** A list of services including 'Division Administration Service', 'Info Mgt Service', 'League Administration Service', 'Player Management Service', 'Player Rating Service', 'Program Development Service', 'Relational Data Org Service', 'Self Service Reporting Service', and 'Team Management Service'. The 'Division Administration Service' is highlighted.
 - Provided Services:** A list of services including 'Player Management Service'. The 'Player Management Service' is highlighted.
 - Buttons:** 'Add Service to Element' and 'Remove Service from Element' buttons are present for both the Required and Provided Services lists.
- Bottom Bar:** Contains 'Help', 'Close', and 'Keep on Top' options.

Now, on to the complicated stuff, Service Provisioning Management.



Service Provisioning with Model Guardian



The key concept here is to determine timely matches between the service needs of the clients and the services provided by the providers.

You first select a service client from the drop down list of service clients. You can also highlight an element in the model and bring it into the service editor. You can then see what services the client requires. Select a required service and you get a list of the providers that provide that service. Lifecycle information for the provider is shown to help you select the best one. You then click one of the "Add to" buttons to either create a new service provision (the connector will be added to the diagram with the selected service as a tagged value and the connector's label), or to add to an existing one (the selected service will be added to the connector's tags and its label). You can click on a service provider to see what services it is providing to the client (provider may provide additional services beyond those of interest to the client for the current scenario). Below that are the lifecycle dates of the client and the provider. You can enter the dates for the connector, i.e. the dates during which the client uses the provider. The date logic will be confirmed before allowing you to save the dates. For example, you cannot enter an actual end date that is prior to the actual begin date.

The last tab is the Deployment Map, which has already been discussed.

There is quite a bit of information captured in a vertical slice diagram. We see...



Service Provisioning with Model Guardian

1. Which services are required to realize the business activity
2. Which services each service element requires and which it provides
3. Which service elements were/are/will be used to satisfy the activity's service requirements
4. Which service elements from lower layers are being used to support those from higher layers as well as when
5. What specific services a service client requires from a service provider from among all those the provider provides as well as when

For a given time period we can easily see anomalies such as...

1. Which service elements will not be available until after the time period begins
2. Which service elements will no longer be available before the time period ends
3. Which service elements require services from providers that will not be available until after services are needed or will become unavailable before the need elapses.

Modeling this in Enterprise is essentially a free-form exercise. There are no checks to see if connections are matching elements according to the service required and provided. There is no checks to see if the lifecycle of the need matches the lifecycle of the providers. There is no date logic for controlling the lifecycle dates of multiple elements and connectors. All this, and more, is handled by the Service Provisioning component of Model Guardian.