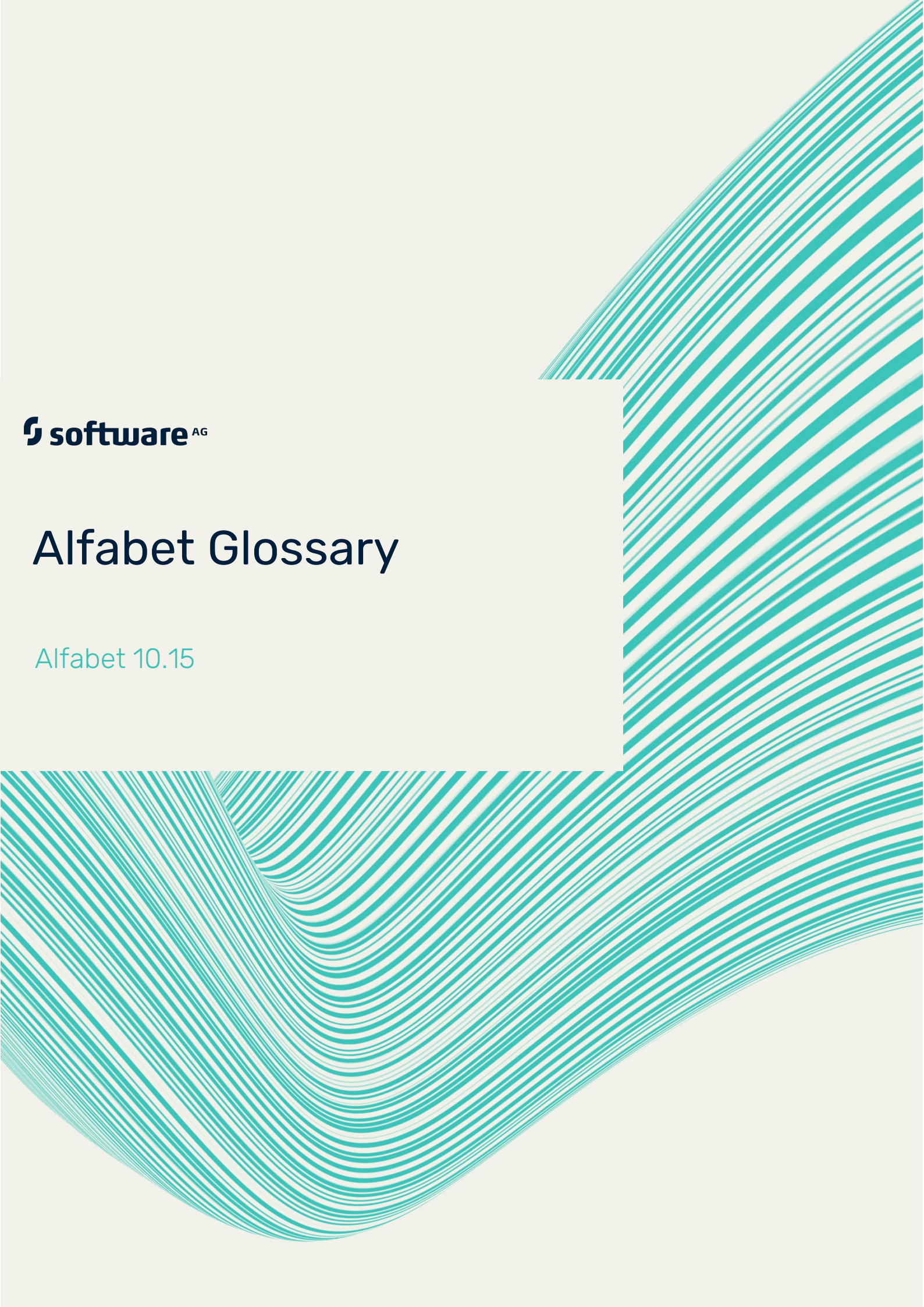

Alfabet Glossary

Alfabet 10.15

Documentation Version Alfabet 10.15.0

This glossary defines terms that are often used in the Alfabet Online Help or when working with Alfabet. If the term is also an Alfabet class, the symbol used in the explorer trees or other places in the Alfabet user interface is also displayed.

A

activity 

An activity is an element in a service diagram that visualizes a task that represents a business process at a granular level. An activity can be directly associated with an agent that performs the task, such as a specified role conducting the business process or an application providing a business service requested to support the task.

An activity typically has a predecessor and successor activity that are connected to that activity through sequence flows. More complex connections between activities are also possible by means of gateways.

An activity can also be connected to an event of the type **Intermediate Event** that occurs while the activity is being carried out. An event of the type **Start Event** typically triggers an activity. An activity can conclude in an event that is of type **End Event**.

activity monitor 

An activity monitor is a type of monitor that alerts subscribed users about changes that have been made to objects in a specified object class. The monitor owner must specify a set of properties that are to be monitored and a set of users that are to be alerted if the monitor is triggered. These users as well as the monitor owner will be informed via email notification if a specified attribute for any object in the class is changed. Monitoring is typically performed in regular intervals over a specific period of time.

ad-hoc milestone

An ad-hoc milestone is a means to plan and track the progress and completion of projects or enterprise releases. Ad-hoc milestones provide complete flexibility in the creation of milestones and can be created independently of a configured milestone template. In contrast to a milestone based on a milestone template, an ad-hoc mileston is created for one explicit project or enterprise release and cannot be reused.

Alfabet query

An Alfabet query is a means to efficiently search and retrieve information in Alfabet by means of object attributes and relationships. Alfabet queries are based on a class-oriented query language and may be configured in a variety of contexts in Alfabet, including full-text searches, configured reports, computation rules, notification monitors, and consistency monitors, for example. Typically, an Alfabet query is defined for a base class. However, it can also be defined to traverse object classes.

As an alternative to the Alfabet query language, native SQL queries may be specified for many of the configurable functionalities in Alfabet.

analytics dashboard

An analytic dashboard is an ad-hoc information-rich data visualization designed by a user based on the embedded third-party tool DevExpress® Dashboard. One or more dashboard items such as charts, scatter charts, grids, cards, gauges, pivots, range maps, treemaps, etc. can be added to the analytics dashboard and filtering options such as combo-boxes, list boxes, and tree views can be leveraged. A configured analytics dashboard data provider is specified for the analytics dashboard in order to fetch the data to display in the analytics dashboard. Analytics dashboards may be shared with other users.

application 

An application is a fully-functional integrated IT product that provides functionality to end users and/or to other applications. As such, an application supports the business to accomplish its mission. Applications operate on a platform made up of hardware and software components necessary to run the application.

The term application is synonymous with application version. An application has a defined lifecycle and may have predecessor and successor versions, thus providing information about the evolution of a specific type of business support or business service.

Applications may possess application variants to account for the need to localize specific aspects of the application, such as the information flows needed to integrate the application in a local architecture environment. From a strategic planning perspective, only the logical structure of applications and the business services and business supports they provide are of relevance.

application architecture

The application architecture is a detailed description of a specific application. The application architecture consists of:

- the application's business architecture, which comprises business supports and business services that the application provides
- the application's information architecture, which comprises the application's interaction with other applications in the enterprise's IT landscapes as well as the business data that are processed or exchanged
- the application's technical architecture, which defines the application's technical platform including its platform tiers and platform layers as well as the technical components that the application requires
- the application's deployment architecture, which identifies the application's technical installations in the enterprise's physical IT landscape

application group 

An application group is a container to logically structure applications in order to view, analyze, and communicate the company's IT architecture. There may be many ways to logically structure applications. Thus, any application may be associated with multiple application groups. Typical ways to group applications include:

- the high-level business processes that applications support
- the organizations that applications support
- the organizations that are responsible for the operation and maintenance of the applications
- the technology that applications use
- ad-hoc assessments of segments of the IT landscape

application variant

An application variant is derived from a base application. An application variant allows for local variations of the base application to be defined including, for example, the interface systems of the information flows, business data, or technical components. The application variant is a descendant of its base application and is nested below it in the explorer. An application variant can be upgraded whenever the base application's definition is modified.

application version

see application

architecture element

An architecture element is an object in the inventory defined to capture the as-is architecture.

archive object

An archive object is a snapshot of a deleted Alfabetobject. When an Alfabet object is archived, a ZIP file is created containing HTML files that display the object profile for the archived object as well as the object profiles of its dependent objects. Each archived object profile displays a preconfigured set of page views, whereby the visibility of these views will depend on the class setting configured for the object class. If a page view displays dependent objects, a user can click the dependent object in the HTML view in order to open another HTML file showing the archived object profile of the selected dependent object.

Alfabet objects are typically archived by a solution administrator. The ZIP file may be downloaded to a local disk and extracted. The relevant HTML file can then be viewed in a browser window. An archive object will be created for each culture setting supported by your enterprise.

artifact

see object

as-is architecture

The as-is architecture comprises all architecture elements that are actively used as well as architecture elements that are planned to be used in the future. The as-is architecture serves as the baseline for the process of planning and proposing new architecture elements for the IT landscape.

In the context of a project, the as-is architecture defines the scope of the IT environment that the project addresses. This is typically the set of architecture elements that is the subject of the change or that may be affected by the changes proposed. These may be architecture elements from the IT landscape such as applications or ICT objects as well as architecture elements from the business landscape such as business processes or organizations. The as-is architecture plays a central role in assessing potential conflicts and redundancies between projects as well as identifying risks to projects.

aspect evaluation

An aspect evaluation is a qualitative assessment of the performance of applications or components assigned to one or more respective application groups or component groups according to defined evaluation criteria. Because an application, for example, may be a member of several different application groups, different qualitative assessments of the same application using the same evaluation criteria are possible for each application group that the application is assigned to.

For example, the component TradeNet may be relevant for different purposes in the enterprise architecture. The component may be suitable to varying degrees both as a business intelligence solution and as an OLAP database. An aspect evaluation would allow you to examine components in a component group based on specific aspect evaluation types, such as business importance or usage. Like conventional evaluation types, aspect evaluation type comprises one or more indicator types that are used to evaluate objects.

An aspect evaluation is based on one or more evaluation types, which are then assigned as aspect evaluation types to the object classes Application or Component. Like conventional evaluation types, aspect evaluation types comprise one or more indicator types that are used to evaluate objects.

Aspect evaluations are currently only available for the object classes Application or Component.

aspect portfolio

An aspect portfolio is a portfolio displaying either a three-dimensional graphic or a BCG quadrant layout that reports qualitative information about applications or components assigned to one or more respective application groups or component groups according to the aspect evaluation and aspect prioritization schemes that have been defined for the aspect portfolio. Aspect portfolios are only available for the classes Application and Component.

aspect prioritization scheme

An aspect prioritization scheme is a prioritization scheme grouping evaluation types in an aspect evaluation in order to assess the performance of applications or components assigned to one or more respective application groups or component groups. An aspect prioritization scheme may also be implemented as an axis or dimension that is used in an aspect portfolio. Aspect prioritization schemes are currently only available for the classes Application and Component.

assignment

An assignment is a task that is defined for a selected object and assigned to a specific user. The assignee is expected to provide the required input for the object by a specified due date. Assignments can be defined to be optional or mandatory. Email notifications may be configured to be sent in the context of the assignment capability.

associated object

An associated object is an architecture element that is associated with and assigned to a domain. Unlike a primary object that is associated with a domain, an associated object is not owned by the domain. For example, an application may be used in different functional contexts and thus associated with several different domains that are not its primary domain. In addition to applications, components, ICT objects, business functions, business objects, business processes, standard platforms, market products, and vendor products may be defined as associated objects.

Whether a domain may have associated objects defined will depend on the solution configuration. If the definition of associated objects is allowed, a domain may have an unlimited number of associated objects, and an object may be assigned to multiple associated domains. However, it is recommended to keep the number of associated domains for each object to a minimum.

association

An association is a connector in a service diagram that displays the relationship between a service diagram element and an architecture element that is referenced by that element. An association is the logical connection from an element in a service diagram to the inventory object. Typically, an association connects a business service requested by a business process or a business function that the requested business service references.

authorized user

An authorized user is the Alfabetuser that has primary responsibility for an object. The authorized user has Read/Write access permissions to the objects that he/she is defined to be the authorized user of. The authorized user is often referred to as the owner of the object and the objects that he/sher owns are commonly referred to as his/her personal objects.

The authorized user may specify another user to be a deputy for any object that he/she owns. Any user that is defined as a member of an authorized user group will automatically have authorization to the objects assigned to the authorized user group.

In addition to authorized users, the concept of roles allows persons and organizations with a functional relationship to an object to be defined.

authorized user group

An authorized user group is a user group that has been assigned to an object. Like the object's authorized user, all users in the authorized user group have Read/Write access permissions to the object. A user group may have an unlimited number of subordinate user groups.

A solution designer may configure the Inheritance and/or propagation of access permissions for authorized user groups. If the inheritance of access permissions has been specified, then all user groups in the user group hierarchy that are descendant to a user group should have the same access permissions to that object. If the propagation of access permissions has been specified, then all user groups in the user group hierarchy that are ascendant to a selected user group should have the same access permissions to that object.

authorized user object

An authorized user object is any object that the user is associated with as the authorized user.

B

base class

The base class is the object class for which a user is seeking results for. For example, a base class is relevant for an alfabet query, native SQL query, or computation rule.

base object

The base object is the object in the inventory that a solution object is based on in the context of solution planning. A solution application, for example, can be based on an application that is part of a project'sas-is architecture. The base object serves as the starting point for the definition of the solution application. All properties and relationships defined for the base application are transferred to the new solution application. The definition of the solution application can then be refined in the context of the solution.

base risk exposure

The base risk exposure is an evaluation of objects according to their potential risks. Each risk object assigned to a risk management group will be assess for its base risk exposure by means of a risk relevance questionnaire, which is based on a configured indicator lookup table. Once an object is assessed for its base risk exposure, risk relevance scores will be computed for that object in order to determine whether the base risk exposure is above a specified threshold. All objects with a score above the threshold should be further assessed for their potential risk damage and risk probability.

blueprint

A blueprint is a master plan or IT strategy that serves as a guideline for planning business support across organizations in the enterprise. In the blueprint planning process, master planners and strategic planners can refer to a blueprint to plan tactical business supports or strategic business supports in business support maps. The blueprint planning process supports the standardization of business support and efficiency in the roll-out of IT support in the enterprise.

bookmark 

A bookmark is pointer to a page view, object profile (or object cockpit), report, or object selected in an explorer. By means of a bookmark, a user can quickly navigate to the relevant view in Alfabet.

Bookmarks can be organized in bookmark folders. A bookmark may only be assigned to one bookmark folder.

bucket

Buckets support recurring budgeting and prioritization on the basis of annual budget allocations. A bucket is a container that allows projects to be structured according to budgets defined for a calendar year. A project may be assigned to multiple buckets and the percentage of the project's costs can be allocated to the buckets. Project scenarios (as well as project solutions) may not be added to a bucket.

business appraisal

Business appraisal is a means for an enterprise to gain an initial understanding of the alignment between the business and the supporting IT. Business appraisals are defined in the context of a business support map. The definition of business supports is not required to conduct a business appraisal.

The business appraisal is an evaluation of the as-is and to-be support for the business process or domain for a selected organization or market product in regard to a specific operational aspect. Once values have been defined for the indicator types configured for the business appraisal, the gaps between the to-be and as-is support can be assessed allowing the enterprise to determine priorities for future projects or its IT strategy in order to align the business and IT.

business architecture

See enterprise business architecture.

business capability

A business capability is an abstract description of what is done in an enterprise to meet its business objectives. For example, Manage Customer Retention is a high-level abstract description of a business capability (what) whereas Send Fax is a concrete description that indicates a specific technology used (how).

Whereas business functions typically focus on a detailed description of business activities, business capabilities view the business activities based on and visualized by a hierarchy of domains (and sub-domains). The business functions are on the leaf level. There is no explicit class for business capability in Alfabet. Domains are used to describe business activities at a coarser level of granularity (Customer Management) whereas business functions describe more concrete activity (Manage Customer Retention).

Guidelines to evaluate an enterprise's business capabilities are defined in a business capability map. The performance of the domains and business functions associated with a business capability can be assessed according to business capability aspects and evaluation types. One or more users can be specified to assess the business capabilities. The assessment allows the organization to determine current performance vs. desired performance and gain a clear understanding of which business capabilities are core to the business and warrant further IT support. The evaluation may be performed from different perspectives. As a result, the business capabilities can be consolidated based on a common understanding of which areas of activity are associated with one another as well as where the gaps and deficiencies in performance lie.

business capability aspect

A business capability aspect represents a specific perspective from which the performance of an enterprise's business capabilities is evaluated.

business capability map

A business capability map allows you to define the guidelines to evaluate the business capabilities defined for the enterprise. The performance of domains and business functions associated with a business capability can be evaluated according to specific business capability aspects and evaluation types by a defined group of evaluators. Mandates are not supported in the context of business capability maps.

business case

The business case is defined for a project and provides a structured approach to estimating the investment and operating costs and benefits associated with a project, and computes standard performance measures such as return on investment or depreciated cash flow.

business data 

Business data represent concrete, logical instances of business objects. Business data are exchanged between applications and their technical components by means of information flows. They are processed in the business services that are provided by the applications/components. A standard CRUD matrix allows users to view and identify conflicts in the business data usage.

business data attribute

A business data attribute specifies characteristics of a business data or business object. A business data attribute will have a type definition (CHAR, BLOB, REFERENCE, etc.) as well as a size specification. The business data attribute defined for a business data will be inherited by the business object that the business data is subordinate to.

business data usage

Business data usage describes whether business data is created, read, update, deleted, or processed in the context of an application, component, information flow, or business service as well as whether business data is an input and/or output of an architecture element.

Business data usage is an important means to identify inconsistencies and redundancies in an enterprise's application landscape. A standard CRUD matrix allows users to view and identify conflicts in business data usage.

business document

A business document is a non-structured document that is often not directly related to IT. They typically represent reports or operating instructions required for performing the business activities associated with a business function. The business document may be specified as an input needed for a business function or as a deliverable generated as output by a business function.

business function 

A business function denotes an atomic business activity that is typically performed in the scope of one or more business processes. A business function is characterized by its independence from the business process context and thus may be applicable across different business processes. At the lowest level, a business process may reference one or more business functions through respective business service requests. In this way, business functions can be understood as lowest-level building blocks for constructing business processes.

The concept of business functions is fundamental to implementing a service-oriented architecture. Business functions standardize business services on the demand-and-supply side by describing the business' IT requirements and offerings in a uniform, formalized, and comparable manner. Additionally, operations can be defined for a business function in order to describe the concrete availability of the associated business services provided by applications or components.

Business functions can also be assigned to either a domain in order to describe the functional structure of the enterprise architecture or to a functional module in order to capture specific business requirements that are or typically can be fulfilled by an application.

For enterprises that structure their business architecture by means of capabilities, business functions are located at the leaf-level of the capability hierarchy and represent a concrete business activity.

business function category 

A business function category bundles and classifies content-specific business functions. The categories are usually derived from business processes.

Business function categories allow you to hierarchically structure the business functions in your enterprise. Each business function may be associated with only one business function category.

business function operation 

An business function operation is a detailed description about how a business service is to be provided by an application in order to fulfill a relevant business service request that is required to realize a business function. An business function operation is defined for the specific business function that it helps to realize. The business function operation defined for a business function is thus assigned to the business services that carry out the business function. A business function can have multiple business function operations defined for it.

Additionally, parameters and return values can be defined for an business function operation in order to describe the interface of a service call as well as provide useful information to drive development activity.

business object 

A business object is the representation of an entity that is relevant to the enterprise's business domain. A business object may represent, for example, a customer, invoice, or order. Business objects can be described with a set of object properties. For example, the business object Customer may have the business data attributes Name and Phone Number.

Business objects are processed by and exchanged between business processes or business functions. A business data is the concrete, logical instantiation of a business object.

business object category 

A business object category bundles and classifies content-specific business objects. Business object categories allow you to hierarchically structure the business objects in your enterprise. Each business object may be associated with only one business object category.

business process

A business process is a set of activities that represent work required to achieve a business objective. Typical business processes include marketing services, selling products, delivering services, distributing products, invoicing for services, and accounting for money received. A business process rarely operates in isolation. Other business processes will depend on it and it will depend on other business processes.

Business processes are structured hierarchically in a business process model. A business process may have as many levels of sub-processes as needed. At the lowest level, a business process may reference one or more business functions by means of business service requests. The alignment of the business and IT is typically planned at a specific level in the business process hierarchy (for example, at the third level in the business process hierarchy).

business process group

A business process group is a container to logically structure business processes. There may be various ways to logically structure business processes. Therefore, any business process may be associated with multiple business process groups.

business process information flow

Similar to a conventional information flow, a business process information flow represents the transmission of information between a source and target. A business process information flow describes the exchange of business objects from one business process to another. A business process information flow establishes an order of succession between the business processes it connects. Consequently, business process information flows typically link business processes that are on the same level in the business process hierarchy.

business process model

A business process model describes a hierarchy of business processes in the enterprise. Multiple business process models may be defined to allow for variations over time or over organizational entities of the enterprise.

business process model version

The business process model version is a version of an existing business process model and serves as a container wrapping one or more related version changes to the business processes in the business process hierarchy. Typically, business process versions capture the concurrent release of changes to the business process in the organization. Because differences in organizational setup, product lines, channels, customer segments, markets or brands might mandate different versions for the same business process being used concurrently, several versions of the same business process may be implemented across the federated organization. Business process versions are planned along with their business supports, which describes the time perspective of the roll out and use of the business process versions.

business process variant

A business process variant is derived from a business process and allows specific stakeholders to analyze business process execution at a more granular level. For example, the high-level business process Sales Order Management can have business process variants defined for the specific process execution flows for the sectors Automobile and Motorcycle. In addition to defining the business services delivered by the business process variant and modeling these in a service diagram, users can also transverse from Alfabet to a specific EPC/BPMN diagram in ARIS or choose a diagram in ARIS and link to the business process variant in Alfabet to study the underlying IT.

business process version

A business process version is a deployed business process specified for a business process model version and represents an activity that is executed in order to achieve a business objective. Typically, business process versions capture the concurrent release of changes to the business process in the organization. Because differences in organizational setup, product lines, channels, customer segments, markets or brands might mandate different versions for the same business process being used concurrently, several versions of the same business process may be implemented across the federated organization.

business question

Business questions allow the focus areas of the business' use cases to be articulated in order to allow the enterprise to capture and evaluate information that supports a particular path of inquiry that is relevant for the business. Business questions are made available for users assigned to a specified user profile. Business questions may be implemented in object cockpits and guide views.

business role

Business roles allow information to be captured about funtional relationships that are relevant to the selected object. Business roles describe funtional relationships that are not fulfilled by people or organizations. For example, actors (such as IT Security Manager, Private Network User, Services Owner), personas (such as Stakeholder, Process Owner, Approver), and job descriptions are typical business roles. Business roles are for informational purposes only and do not impact access permissions. Object class stereotypes can be configured for the class **Business Role**.

business service 

A business service is an IT service that can be provided by an application, component, local component, organization, market product, business process, ICT object, or solution building block in order to realize a specific business function. Depending on the solution configuration, applications and components may provide one or more business services for the same business function. A business service thus fulfills a specific business service request that is posed by a business process for that specific business function.

Business services operate on business data by means of the applications, components, organizations, market products, or business processes that provide the service. Business services may also operate on

business objects by means of the ICT objects or solution building blocks that provide the service. An operation can be defined to describe how a business service is to be provided by the providing object.

A business service inherits the authorized user and/or authorized user groups from the object that provides it.

business service request 

A business services request formulates the need of a business process or market product for IT support in the form of a business service based on a business function. Based on this information, requests can be matched with concrete business services that are provided by applications or components.

The business service inherits its authorized user and/or authorized user group from the application or component that provides it.

business support

A business support defines an object such as an application that provides support to the enterprise to carry out its business. Business supports typically provide support to business processes, although an enterprise may describe its business support to domains instead. Furthermore, business supports are typically executed by organizations responsible for the business processes/domains. Some enterprises may specify market products instead of organizations. Alfabet supports the definition and analysis of operational business supports, solution business supports, strategic business supports, and tactical business supports.

business support map

A business support map is the graphic visualization of the business support provided by specified objects in order to support the planning of the short-term, medium term, and long-term to-be architecture. The map is visualized as a matrix with an X-axis and a Y-axis. The X-axis of a standard business support map will display business processes and the Y-axis will display organizations that are supported. In some industry segments, it is more relevant to analyze the business support for market products than for organizations. If this is the case, market products may be configured for the Y-dimension of the business support. Furthermore, some enterprises may describe business supports to provide support to domains of the business rather than business processes. If this is the case, domains may be configured for the X-dimension of the business support.

The matrix cells consist of the business supports that support the corresponding X-axis object and Y-axis object. In the context of master plans or IT strategies, business support matrices may display operational business supports, tactical business supports, strategic business supports. In the case of a solution map, solution business supports will be displayed.

A business support map may be specified for an IT strategy or master plan that is specified as a blueprint. The definition of the business support map for a blueprint strategy or master plan may be embedded in another business support map to serve as a guideline for planning and standardizing business support across organizations when planning the enterprise's business architecture.

Alfabet supports the configuration of customized business support matrices to allow large numbers of business supports to be simultaneously captured for a specified set of organizations/market products or business processes/domains.

C

class

See object class.

class-based risk management template 

A class-based risk management template allows you to specify the object class that is the target of a risk assessment. For each object class that is being assessed for risk, a specific indicator lookup table can be defined capturing the questionnaire items and risk relevance score.

The settings defined for the ascendant risk management template, including the risk damage indicator type, risk probability indicator type, and risk portfolio will be inherited by all risk objects that belong to a risk object group referencing the risk management template.

clustering

Clustering is an association between a physical device and a logical device. By means of clustering, a physical device can be assigned to a logical device, whereby the logical device serves as a kind of logical cluster for the physical device. For example, a virtual machine could be considered a logical cluster of physical devices.

A logical device can cluster a multiple number of physical devices and a physical device can be clustered by a multiple number of logical devices.

collaboration topic 

A collaboration topic is a free-form exchange of information and ideas about a specific object or view in Alfabet by members of the user community. A collaboration topic can be created for an object in an object profile/object cockpit or a specific page view and all invited users can add a post to the collaboration topic. Any user with ReadOnly and ReadWrite access permissions to the object view or page view can start a collaboration topic.

color rule

A color rule is based on one or more Alfabet queries or native SQL queries that are configured to color a found set of objects in standard Alfabetbusiness support maps and diagrams. If the color rule functionality is activated for a business support map, all activated color rules will be executed and the matrix cells colored accordingly. In order to visualize color rules in diagrams, color rules must be assigned to a diagram view. When the diagram view is selected in a diagram, all activated color rules will be executed and the diagram objects will be colored accordingly. The color rules are automatically added to legends where they are applicable.

compliance control 

A compliance control represents a critical question of inquiry that must be answered to assess the conformity of a specific set of objects to regulatory standards. Compliance controls are bundled in a compliance control set and are hierarchically structured with one top-level control and an unlimited number of levels of sub-controls. The hierarchy of compliance controls constitute the series of compliance controls that an object must pass through in order to reach the leaf-level compliance control. The leaf-level compliance control in the compliance control hierarchy entails the question that is posed in the compliance project.

One or more compliance policies defined for the ascendant compliance control set may be assigned to the compliance control. Once a compliance project is running, the compliance controls are instantiated as compliance project controls in the context of the compliance project.

compliance control set 

A compliance control set serves as a blueprint that can be used for multiple compliance projects. It typically represents a regulatory compliance standard and is structured as a hierarchical catalog of questions (such as COBIT or SOX) defined to evaluate specific objects in an enterprise.

The compliance control set bundles compliance controls, which are hierarchically structured with one top-level control and an unlimited number of levels of sub-controls. A compliance control at the lowest level of a leaf in the hierarchy entails a question that is posed about the target objects.

One or more compliance policies can be defined for a compliance control set. A compliance policy defines the rules to find the target objects as well as the persons responsible for answering the questions about the target objects.

The compliance control set also specifies the indicator type that is to be used as the standard metric when the target objects are evaluated. In others words, the indicator type comprises the values that users can select to answer the question posed about the target objects. Only one indicator type may be defined for the compliance control set.

compliance domain 

A compliance domain represents part of an enterprise that serves as the valid area for which a compliance project should be executed. For example, the compliance domain could be a geographic area or a specific topic.

compliance policy

A compliance policy is created for a compliance control set and defines which target objects are to be evaluated in the context of a compliance project as well as the persons to perform the evaluation. This is achieved by defining rules to find the objects that are the target of the compliance project as well as the persons responsible for answering the questions regarding the architecture elements. The persons identified to answer the questions could be the authorized users of the target objects, users having roles defined for the target objects, or users who are not directly associated with the target objects. The target objects as well as responsible users who are not directly associated with the target objects are found by means of queries.

A different compliance policy must be defined for each object class that is targeted by the compliance project. A compliance control set may have an unlimited number of compliance policies. Once all compliance policies are created for the compliance control set, the compliance policies must be assigned to the compliance control that they are relevant for. The assignment of the compliance policy to the compliance control determines which objects must be evaluated for the question posed by the compliance control.

compliance project 

A compliance project represents an inquiry launched in an enterprise for a regulatory evaluation of a specific set of objects. A compliance project is based on a compliance control set and compliance domain. The compliance control set defines the objects targeted by the compliance project, the questions to be asked, and the persons who are responsible for answering the questions.

A compliance project serves as an instantiation of the compliance control set for a specific time and area of validity that is determined by the compliance domain. For example, a compliance project based on a compliance control set representing SOX might be the SOX assessment in Q1/2007 for a regional subsidiary in the enterprise. A compliance project can only be based on a compliance control set if the release status of the compliance control set has been set to compliance domain.

A compliance control set may have an unlimited number of compliance projects that represent various inquiries across time or regions, but only one compliance project may be active for a selected compliance domain and defined date. The compliance project inherits all compliance policies defined for the compliance control set it is based on.

An active compliance project indicates that the release status of the compliance project has been set to **Active** and will begin on the start date defined for the compliance project. Once the start date has been reached, the compliance project is considered to be running.

compliance project control 

A compliance project control represents a critical question of inquiry that must be answered in the compliance project. Compliance project controls are defined in the compliance control set as compliance controls and become instantiated once the compliance project's release status is set to **Active**. The compliance project control at the leaf-level of the compliance project control hierarchy entails the questions posed about the target objects in the compliance project.

component 

A component is a reusable block of functionality that is implemented by either hardware or software. In Alfabet, components typically represent:

- An interface for data exchange (see information flow)
- A shared element that, when deployed, is able to deliver business services to multiple applications (for example, a database)
- A subsystem of an application that is managed as a separate entity with its own independent lifecycle
- A semi-finished application such as an application that requires configuration to be functional and is thus used in different configurations (for example, a specific packaging of SAP CO may be used as the basis for several applications that have different configurations)

Usually, a component does not deliver functionality to end users like a business service does. However, a local component may be employed in a specific application context to support an application that provides business services to users. Typically, however, the local component provides technical services in the context of an application.

In contrast to local components, standard components may also be developed, managed, and operated stand-alone, independent of the applications that make use of them. In Alfabet, standard components are referred to simply as components. Similar to applications, standard components operate on a platform made up of hardware and software components (local components) necessary to run the component. For components of the type Service, technical services can be defined in order to fulfill technical needs that are necessary to support business service request.

A component may reference a vendor product, thus indicating that it has been derived from the vendor product. Components can be structured in a component group, component category, and/or component catalog.

component, local

See local component.

component catalog 

A component catalog provides an overview of the level of standardization of components for a defined period of time. In this way, an enterprise can determine if a component is approved as a standard component, has limited support, or is not permitted for future use, etc. In Alfabet, both the as-is architecture and proposed architecture changes can be analyzed and verified for compliance, thus supporting the technical architecture to evolve in alignment with the enterprise's strategy.

A component catalog typically refers to functional, geographical, or organizational sub-entities of the enterprise. A component can belong to more than one catalog and have different lifecycles in different catalogs.

Any component assigned to a component catalog is considered a component catalog element. Every component catalog element has a lifecycle independent of its associated component. The lifecycle indicates when the component is defined as standard in that catalog's scope. If a component is not listed in a specific component catalog, the component is considered to be available to the entire corporation according to its lifecycle definition.

component catalog element

A component catalog element is a constituent of a component catalog. Each element is associated with one standard component in the enterprise. A component catalog element may be assigned to multiple catalogs.

Each component catalog element has a lifecycle independent of its associated component. By means of a component catalog element, a component may have a different lifecycle defined for each catalog. The lifecycle indicates when the component is defined as standard in that catalog's scope.

component category 

A component category bundles and classifies content-specific components. Component categories allow you to hierarchically structure the components in your enterprise. Each component may be associated with only one component category.

component group 

A component group is a container to logically structure components in order to view, analyze, and communicate the company's IT standards. There may be various ways to logically structure components. Thus, any component may be associated with multiple component groups, allowing the same component to be considered in different semantic contexts.

For example, SAP BW is typically considered a member of a component group Business Intelligence Tools. However, it could also be a member of the groups Reporting Tools and OLAP Tools.

component module

A component module is a means to logically group local components for a selected application or standard component. A local component may be assigned to multiple component modules. The business, technical, and information architectures for the component module are an aggregation of the business, technical, and information architecture of the local components bundled in the component module.

component test

A component test is a specified plan to test a selected object and document the results of the test once executed. Component tests are configurable for the following object classes: Application, Component, Component Module, Local Component, Deployment, Device, Standard Platform, and Vendor Product.

computation rule

A computation rule allows numeric indicators to be automatically calculated for objects in a specific object class. In order to define the computation logic, the user must define a base class and an associated attribute of the type `Integer`, `Real`, `Reference`, or `ReferenceArray` that is to be computed. The indicator type to be calculated must be assigned to the computation rule. A scaling scheme can be applied to the computation rule, if needed.

The following can be defined by means of a computation rule:

connection data format

A connection data format describes the data format used for the transfer of business data via a specific information flow. Examples include ASCII, XML.

connection frequency

A connection frequency describes how often a specific information flow is used to transfer business data between the two associated applications or their respective components. Examples include daily, monthly.

connection method

A connection method describes the method of transfer used by a specific information flow to transfer business data between the two associated applications or their respective components. Examples include TCP/IP, file transfer, message queue.

connection type

A connection type describes the mode of transfer used by a specific information flow to transfer business data between the two associated applications or their respective components. Examples include batch and online.

consistency monitor 

A consistency monitor is a type of monitor that supports the system-wide maintenance of objects in the Alfabet database. The consistency monitor is specified to periodically search for inconsistencies among objects. Each consistency monitor is based on an alfabet query or native SQL query that defines the object classes targeted by the query as well as the inconsistent properties to be detected. If an inconsistency is found by the query, the monitor alerts the authorized user of the object via an assignment about the inconsistency. The timely completion of the assignment triggered by a consistency monitor can be tracked by the solution administrator.

consumed business service

A consumed business service is a business service provided by an application that is consumed by an organization to fulfill a business service request of a business process.

contract

A contract is a legal document that stipulates the terms of agreement between organizations buying products and services and organizations or vendors providing the products and services.

A contract may be referenced as a master contract by multiple contracts.

Contracts are structured in contract folders. A contract can be assigned to only one contract folder.

The enterprise may configure object class stereotypes for the class Contract. For example, one stereotype for the class Contract might be Software Licenses and another object class stereotype could be Service Contracts.

contract deliverable

A contract deliverable is a specified architecture element or set of resources that must be provided in order to meet the terms of the associated contract or contract item. Object class stereotypes may be configured for the class **Contract Deliverable**.

contract group

A contract group is a container that allows contracts to be structured. There may be various ways to logically structure contracts. Thus, any contract may be associated with multiple contract groups, allowing the same contract to be considered in different contractual contexts.

contract item

A contract item is a part of a contract that a specified organization is typically responsible for maintaining or acting upon. For example, a contract item could be the license purchase of a software product and another contract item could be the maintenance of the software product.

Multiple contract items can be defined for a contract. The responsible organization as well as the cost information can be captured for a contract item. A contract item may also have contract deliverables that specify the architecture elements and resources to be provided in order to fulfill the contract agreement.

The enterprise may configure object class stereotypes for the class Contract Item. For example, one stereotype for the class Contract Item might be License Purchase and another object class stereotype could be Maintenance Agreement.

cost

A cost represents a monetary value required for a specific type of activity or service for an object in the IT landscape for a given period of time.

cost center 

A cost center is a means to centrally define costs for a specified period of time and allocate them to a group of architecture objects (applications, deployments, ICT objects, projects, or service products) according to a defined allocation scheme.

A cost center is based on a cost center type, which serves as a template to define cost centers. The costs assigned to a cost center are based on the cost types configured for the associated cost center type. A cost center can be assigned to an unlimited number of cost center groups.

cost center group 

A cost center group is a container to group and hierarchically structure cost centers. A cost center group can be assigned an unlimited number of sub-groups as well as an unlimited number of hierarchy levels in the tree structure. A cost center group may be subordinate to only one ascendant cost center group.

Each cost center group may have an unlimited number of cost centers assigned to it. A cost center can be assigned to an unlimited number of cost center groups.

cost center type

A cost center type serves as a template for creating cost centers. One or more cost types can be assigned as standard cost types to a cost center type. When a cost center is created, it is based on a cost center type. The cost types assigned to the cost center type will be automatically assigned to the cost center.

cost type

A cost type is a classification of costs. Cost types are defined by the enterprise to ensure comparability in the definition and evaluation of the investment and operation costs associated with ICT objects, applications, deployments, projects, and service products. Cost types are critical for the definition of business cases as well as for cost planning for projects. An unlimited number of subordinate cost types may be defined for any cost type in order to allow for bottom-up cost estimation and analysis.

CRUD matrix

A CRUD matrix is a well-established means to represent business data usage for applications, components, or business services. CRUD refers to the four major interactions (Create, Read, Update, and Delete) between architecture elements and business data. Furthermore, the letters I and O (Input and Output) indicate whether the business data are being used in the incoming or outgoing information flows defined for the applications. A variety of analyses using a CRUD matrix help the user to identify inconsistencies and redundancies in the way business data are used in the IT landscape.

In a CRUD matrix, architecture elements represent one dimension of the matrix and business data represent the other dimension of the matrix. The corresponding cells are filled with a combination of the letters C,R,U, D, I, nd O thus indicating if and how an architecture element and business data are related to one another.

culture

A culture constitutes the base configuration of the default primary language displayed on the user interface when the user first logs in as well as the date, time, and number formats, etc. used to capture and render dates, times and numbers in the Alfabet user interface.

custom property

A custom property  is an object class property that has been configured in order to capture enterprise-specific data for a specified object class.

Multiple custom properties can be created for each object class. Like private properties, custom properties (also referred to as public properties) can be excluded from a class setting and therefore be hidden for users accessing Alfabet with the associated user profiles. Custom properties can also be included in the definition of a class key requiring users to enter unique values for a set of standard and custom properties when defining objects in the object class.

In order for users in the user community to define values for the custom properties, the properties must be associated with data entry fields in custom editors. Custom properties can be displayed in object profiles and object cockpits. Additionally, custom properties can be used to search for objects in Alfabet's search functionalities and can be specified in alfabet queries and native SQL queries that are used in, for example, configured reports.

D

data retention policy

A data retention policy supports the storage and management of business data. Data retention policies provide a means to document standard information about how business data should be retained and stored including the amount of time that business data should be retained, rules for archiving the business data, permissible means to store and access the business data, and the levels permissible to encrypt the business data.

date monitor 

A date monitor is a type of monitor that alerts subscribed users about approaching dates that have been defined for objects in a specified object class (for example, the approach of an object's start or end date). Two kinds of date monitors are available in Alfabet: object-specific date monitors and system-wide date monitors that target all objects in a specified object class.

Object-specific date monitors are defined by an individual user to keep track of the approach of a date for specified objects. The monitor owner must specify a set of properties that are to be monitored and a set of users that are to be alerted if the monitor is triggered. These users as well as the monitor owner are informed when the target date approaches for the specified attribute and can then decide upon the appropriate action to take. Monitoring is performed in regular intervals over a specified period of time.

In addition to the conventional Alfabet date monitors, system-wide defined date monitors can be configured for an object class. When a specified date approaches, all authorized users responsible for objects in the relevant object class will be sent notifications per email asking them to review the objects that they are responsible for.

demand 

A demand is a request for change in the enterprise's IT landscape. Demands in Alfabet are typically strategic or impact the IT landscape and therefore must be taken into consideration in the enterprise's project planning process.

Typically, a demand is mapped to a number of architecture elements that need to be changed in order to fulfill the demand. This information can be used to initiate the architecture planning process as well as to identify synergies and redundancies across demands. Once a demand is identified, it can be assigned to and realized via a project. A demand can only be assigned to one project.

demand group 

A demand group is a container to logically structure demands. There may be various ways to logically structure demands. Therefore, any demand may be associated with multiple demand groups.

deployment

A deployment is the logical set of installed elements that constitute one instantiation of an application or component. A deployment element identifies the device that the respective architecture element is installed on for each of the installed architecture elements including the application or component itself and all elements of that application's or component's stack.

deployment element

A deployment element is a constituent of a deployment. Every deployment element is either directly associated with a device, references another deployment element through reuse, or employs local components directly. A deployment element may be associated with a stack element for the stack that is deployed by the deployment.

deputy

A deputy is an Alfabetuser granted Read/Write access permissions to an object in order to act on behalf of the authorized user.

device 

A device represents either real hardware or virtual hardware that applications and components can be deployed on. Devices are associated with deployment elements and are found in locations. A device may be defined as a physical device or a logical device and may be the source of a network route.

device group 

A device group is a container to logically structure devices in order to view, analyze, and communicate the company's physical IT infrastructure. There may be various ways to logically structure devices. Therefore, any device may be associated with multiple device groups.

diagram view item

A diagram view item is the configuration that specifies which information should be displayed about the objects visualized for a selected diagram view in a Alfabet diagram. The diagram view item captures the

attributes and indicators that should be displayed for the object. The specified attributes and indicators are then displayed on the objects in the diagrams, which are usually visualized as rectangles or lines, if the objects are a connection item such as information flows, roles, or rules.

diagram view

A diagram view is a configuration that is associated with a diagram. It allows users to superimpose qualitative information - such as aggregated indicators or attribute values - associated with these architectural elements. Diagram views can be implemented in diagrams displaying applications, business processes, devices, domains, frameworks, platforms, and solution building blocks. Multiple diagram views may be defined and made available for a diagram. A diagram view may be reused for multiple diagrams.

discussion group 

A discussion group is a preconfigured group of users that have been defined permission to discuss objects in specified object classes.

A discussion group must have at least one subordinate class-based discussion group defined, which specifies the object class that is the focus of discussion, the discussion statuses to be implemented for the discussion, whether members of the discussion group have been granted Read/Write or ReadOnly access permission to the objects being discussed, and whether automatic email notifications should be sent in the context of discussion activity.

Other discussion groups may be invited to join a discussion that has been initiated by the discussion group managing the discussion. In this case, the discussion group that is to be invited must be included in the discussion group's set of manageable discussion groups and must have a relevant class-based discussion group defined for it. For example, if the managing discussion group is discussing an application and invites another discussion group to provide input to the discussion, the invited discussion group must also have a class-based discussion group defined for the object class Application.

domain 

A domain represents a functional entity in the domain model and allows the enterprise architecture to be hierarchically partitioned into disjoint segments and structured from a functional or technological point of view, for example. A domain is based on an object class stereotype which represents a specific level in the domain hierarchy.

One of the goals to achieve an effective service-oriented architecture is to structure the IT landscape so that each architecture element (business functions, business objects, business processes, ICT objects, applications, components, standard platforms, and vendor products) is owned by only one domain, which defines the principal functional context for the object. Each architecture element may only be owned by only one domain, which defines the principal functional context for the object. In some cases, a domain will have associated objects. These are objects that are relevant for the domain in a functional context even though the domain is not the owner.

For enterprises that define their business architecture by means of capabilities, the business capability references a domain in the domain model. In the context of Capability Management, a domain describes

business activities at a coarser level of granularity whereas the business functions, located at the leaf level in the domain hierarchy, describe concrete activity.

domain glossary

A domain glossary allows users to define and search for domain-specific terminology defined for objects that are assigned to a domain. Domain-specific terminology includes aliases and descriptions of the objects.

domain group 

A domain group is a container to logically structure domains. There may be various ways to logically structure domains. Therefore, any domain may be associated with multiple domain groups.

domain model

A domain model describes a hierarchy of domains that are based on configured domain stereotypes. Multiple domain models each with a configured number of levels in the domain hierarchy may be configured for the enterprise in order to allow for various functional entities to be described and assessed in the context of a service-oriented architecture.

domain stereotype

A domain stereotype is an object class stereotype created for the object class `Domain` in order to represent a classification of a functional entity in the domain model. Each domain created in Alfabet is based on a domain stereotype.

E

enterprise application architecture

An enterprise application architecture defines the application view of the enterprise architecture and is described in terms of the enterprise's IT applications.

enterprise architecture

An enterprise architecture is a comprehensive method and framework used to manage and align an enterprise's business processes, organizations, information technology (IT), software and hardware, local and wide area networks, people, operations, and projects with the enterprise's overall strategy.

enterprise business architecture

An enterprise business architecture defines the business view of the enterprise architecture and describes the enterprise's business processes, business functions, market products, and organizations.

enterprise calendar

An enterprise calendar is a calendar created for the enterprise that contains specified days that are blocked out as holidays or weekends. Multiple enterprise calendars can be created for all relevant regions of the enterprise. Users can import an enterprise calendar as a template when creating their own personal calendars.

enterprise deployment architecture

An enterprise deployment architecture defines the deployment view of the enterprise architecture and describes the enterprise's locations, devices, and deployments.

enterprise information architecture

An enterprise information architecture defines the information view of the enterprise architecture and describes the enterprise's business objects and associated standards.

enterprise release

An enterprise release is the process in which changes to the enterprise architecture are planned and managed via a governed process. For each enterprise release, a set of milestones are specified that constitute the stage gates for the approval and execution of the release and thus allow the enterprise release cycle to be managed.

A set of release statuses can be configured for the object class Enterprise Release in order to specify the statuses of the release cycle (for example, New, Planned, In Execution, Closed). Workflows can be configured to manage the different stages of the release cycle.

The enterprise release consists of one or more enterprise release items that are typically based on projects that deliver architectural change to the enterprise. A set of release statuses can also be configured for the object class Enterprise Release Item in order to specify the stages of project registration, execution, and completion.

enterprise release item

An enterprise release item expresses the deliverables that will be provided for an enterprise release. Typically, an enterprise release item is based on a project that has been defined to provide the architectural change. Alternatively, enterprise release items can be based on applications, components, or standard platforms. In this case, the architecture element constitutes the deliverable to the enterprise release.

A set of release statuses can also be configured for the object class Enterprise Release Item in order to specify the stages of project registration, execution, and completion.

enterprise technical architecture

An enterprise technical architecture defines the technical view of the enterprise architecture and describes the enterprise's software and hardware components, infrastructure, and associated standards, such as component catalogs, standard platforms, and master platforms.

evaluation

An evaluation is a set of values that measure the performance of an object with respect to a specific dimension. An evaluation is based on an evaluation type that comprises a bundle of indicators which are the calculated output values of preconfigured indicator types.

Typically, the authorized user or authorized user group for an object conducts the evaluation.

evaluation type

An evaluation type bundles one or more indicator types for use in the evaluation of a specific dimension of an object's performance. Users define an indicator for each indicator type bundled in the evaluation. For example, typical evaluation types could be Complexity and Standardization Status. The evaluation type Complexity might have the indicator types Number of Interfaces, Number of Modules, and State of the Art, and the evaluation type Standardization Status might have one indicator type also named Standardization Status.

An evaluation type may be configured once and reused for multiple object classes. In this way, all system-related classes such as Application, Component, and Standard Platform, for example, could be evaluated for their complexity, standardization, etc.

Evaluation types can also be grouped into prioritization schemes that, in turn, may be implemented as axes in portfolios.

Access to the evaluation types used in Alfabet evaluations can be controlled via user profiles. A user can view any evaluation type that has either no user profile defined or the same user profile as that which he/she is currently logged in with.

event

An event is something that happens before, during, or after a business process is conducted. An event can be set in motion by triggers such as a message, rule, or error. An event can be connected to an activity through connectors such as sequence flows.

There are three different types of events:

- A start event is an event that happens before a business process is conducted. A start event triggers a business process and is connected to the first activity of a service diagram, typically by means of a sequence flow.
- An intermediate event is an event that happens while a business process is being conducted. An intermediate event is typically situated between activities and connected to them by means of service diagram connectors.
- An end event is an event that happens after a business process is conducted and the business process outcome is achieved. An end event is connected to the last activity of a service diagram, typically by means of a sequence flow.

express view

An express view is a link to a specific Alfabet view or explorer displaying data that allows individuals inside or outside of the Alfabetuser community to view Alfabet information. When the express view is created, an email notification is automatically mailed to the specified recipient who receives a URL that allows him/her to access the current page view in Alfabet.

F

federated architecture

A federated architecture is an architecture that is typically managed in silos. Such an architecture allows a holding company to structure and communicate the elements that are shared across the enterprise, the elements that are common to some but not all organizations in the enterprise, and the elements that are exclusive to a specific organization. The assignment of mandates to Alfabetobjects allows the holding company to structure the objects in the enterprise architecture and regulate their visibility to only those users who should see them.

framework

A framework is a means to define a semantic structure for objects managed in Alfabet, regardless of the object class that an object belongs to. In Alfabet, a framework is a hierarchically structured collection of framework groups in which a user can navigate through the levels of the framework structure.

framework group

A framework group is a container to structure objects from different object classes in order to view, analyze, and communicate the company's enterprise architecture.

functional module 

A functional module is assigned to a domain and bundles business functions to capture the specific business requirements that are or typically can be fulfilled by an application. In this sense, a functional module can be considered an application prototype that captures functional aspects necessary to the enterprise architecture. A business function can be assigned to multiple functional modules.

G

gateway 

A gateway is a connector in a service diagram that controls the convergence and divergence of the sequence flows. A gateway is typically placed between two activities whose dependencies cannot be expressed through simple connections. Gateways can also be combined to visualize even more complex dependencies between activities.

Three types of gateways are supported:

- AND gateways connect an activity with a set of activities that need to be conducted simultaneously after the predecessor activity. It visualizes the logical AND operator for the successor activities.
- OR gateways connect an activity with a set of optional activities that can be conducted after the predecessor activity. It visualizes the logical OR operator for the successor activities. This means that a some of the successor activities are conducted. However, in contrast to the AND gateway, not all of them are necessarily conducted.
- XOR gateways connect an activity with a set of competing activities, one of which must be conducted after the predecessor activity. It visualizes the logical XOR operator for the successor activities. This means that only one of the successor activities is conducted due to a certain condition. An XOR gateway is typically used to visualize a decision tree with a yes/no answer. The choice of the successor activity is dependent on the answer to the question represented by the XOR gateway.

generic attribute

A generic attribute allows the ad-hoc capture of content in a semi-structured form for a specific application, component, deployment, device, standard platform, stack, deployment element, standard platform element, stack element, or stack configuration item. A generic attribute can be used if one or more attributes are required for informational purposes only and each attribute is only used for a small subset of

objects rather than for all objects in an object class. A generic attribute is an alternative to configuring a custom property for an object class.

generic reference data

A generic reference data is an object representing a classification and is an alternative to enumerations to capture custom data. Typical classifications that already exist in Alfabet are the reference data types connection types, connection methods, connection frequencies, and connection data formats.

guide page

A guide page is a means to configure a start page with a predefined layout of elements. The guide page configuration has been replaced by the more flexible guide view configuration and is available for backward compatibility.

guide view

A guide view is a configured start page for Alfabet that has the look-and-feel of and Internet Web page. The guide view provides users with links to Alfabet functionalities and documents, embedded reports, pictures, and informational text. Guide views serve as an alternative to the standard start page provided by Software AG.

H

I

icon gallery

An icon gallery is a configurable collection of icons that can be used to graphically visualize the quantitative value of indicators in evaluations and diagrams. An icon gallery is typically a set of mutually related images whereby each icon represents a specific value range. For example, the three stages of a traffic light signal could represent 3 different indicators.

ICT object

An ICT object (ICT = Information and Communication Technology) is an abstract object that can represent either an architecture element (for example, an application, solution building block, component, device,

standard platform, or vendor product) regardless of its versioning, or multiple architecture elements that are related for business or financial reasons.

An ICT object is owned by an organization that is usually responsible for the budget of the architecture elements assigned to the ICT object. As such, ICT objects are key to enterprise strategy and master planning. The use of ICT objects is advantageous in that the planner must not initially commit him/herself to a certain version of, for example, an application. Later, at the stage of detailed planning, the ICT object can be replaced by a specific concrete version.

ICT objects can be structured in an ICT object category as well as an ICT object group.

Object class stereotypes may be configured for the class **ICT Object**.

ICT object category 

An ICT object category bundles and classifies content-specific ICT objects. ICT object categories allow you to hierarchically structure the ICT objects in your enterprise. Each ICT object may be associated with only one ICT object category.

ICT object catalog element

An ICT object catalog element is a constituent of a component catalog. Each element is associated with one ICT object in the enterprise. An ICT object catalog element may be assigned to multiple component catalogs.

Each ICT object catalog element has a lifecycle independent of its associated ICT object. By means of an ICT object catalog element, an ICT object may have a different lifecycle defined for each component catalog. The lifecycle indicates when the ICT object is defined as standard in that catalog's scope.

ICT object group 

An ICT object group is a container to logically structure ICT object in order to view, analyze, and communicate the main building blocks of the company's IT landscapes. There may be various ways to logically structure ICT objects. An ICT object may be associated with multiple ICT object groups, which allows the same ICT object to be considered in different semantic contexts.

request ICT object

A request ICT object is an object that is typically created and used in the context of a workflow in which a request for a new ICT object is defined. The request ICT object can be associated with the applicationss, components, devices, standard platforms, and vendor products that the requested ICT object would represent. The process of defining a request ICT object is for documentation purposes only. A request ICT object does not automatically trigger an ICT object. The definition of a new ICT object typically occurs in the context of a demand. Object class stereotypes may be configured for the class **Request ICT Object**.

inactivity monitor 

An inactivity monitor is a type of monitor that alerts subscribed users about the absence of activity occurring to objects in a specified object class. The monitor owner must specify a set of objects that are to be monitored and a set of users that are defined as listeners to be alerted if the monitor is triggered. The monitor owner is typically the user responsible for the objects defined for the monitor. The monitor owner as well as the listeners will be informed via email notification if a specified object has not been edited or reviewed within a specified period of time. If the monitored object does not need to be changed, the monitor owner can mark the object as reviewed.

income

Income represents a monetary value expected to be earned in a given period of time from an activity or service related to an object in the IT landscape. Income is key to the definition of a project'sbusiness case and serves as the basis for the analysis of return on invest.

income type

An income type is a classification of income. Income types typically represent benefits in a business case and are defined by the enterprise to ensure comparability in the monetary evaluation of activities and services for projects. Income types are critical for the definition of business cases in Alfabet.

indicator

An indicator is the qualitative or quantitative measurement of an object's performance. Indicators may be entered manually, by means of an external system interfacing with Alfabet, or computed or aggregated within Alfabet. Each indicator is based on an indicator type that defines the dimension and scale of measurement. One or more indicator types are bundled in an evaluation type and may be implemented in evaluations, prioritization schemes, and portfolios.

indicator lookup table 

An indicator lookup table is a means to evaluate the base risk exposure of a specified set of objects and determine whether the objects should be assessed for risk. The indicator lookup table captures two different evaluation types used to evaluate the base risk exposure. One evaluation type groups a set of indicator types that represent the questions to be answered in order to determine whether an object should be assessed for risk. The other evaluation type groups a set of indicator types that allow the answer to each question to be translated to a risk relevance score - a score that determines how high the risk is to the object and whether the object should/should not be channeled to the next stage of risk assessment. The indicator lookup table thus also specifies the mapping between the answers to the questions and the risk relevance score. Each potential value defined for a question will have a corresponding risk relevance score value.

indicator type

An indicator type defines a dimension of measurement regarding the performance of an object in the IT landscape. The indicator is the value defined for an indicator type in the context of an object in the IT landscape.

A single indicator type or multiple indicator types can be bundled in an evaluation type that is used to evaluate an object. For example, the indicator types Number of Interfaces, Number of Modules, and State of the Art could be assigned to the evaluation type Complexity, and the indicator type Standardization Status could be assigned to the evaluation type also named Standardization Status.

Each evaluation type may have one indicator or more assigned to it. An indicator can be either:

- manually entered by a user
- selected from a predefined set of values by a user
- calculated by Alfabet according to defined computation rules
- calculated based on customer-specific code provided by Alfabet
- imported via an interfacing system such as ADIF

Indicator types having a predefined set of values may be associated with an icon gallery, allowing for indicators to be visualized in diagram views as well as other configured reports by means of an icon rather than a numeric value.

information architecture

See enterprise information architecture

information flow 

An information flow describes the exchange of business data between source and target applications, components, devices, or peripherals. The interface logic that is required for the exchange of information is an integral part of the information flow. Information flows between applications are realized through concrete interface systems which are typically embedded in the application's platform as a platform component. Within an application's platform, a local component can also be specified as a source or target interface enabling the data transfer of the information flow. An information flow may possess source and target interfaces as well as interface systems. Information flows can be classified by the connection type, connection method, connection frequency of the information exchange, and the connection data format describing the data exchange.

instance

See object

interface

An interface is the technical component associated with an information flow that realizes the technical compilation necessary for the data exchange. The source and target interfaces must be defined as components or local components in the platform of the respective application. An information flow may possess source and target interfaces as well as interface system.

interface system

An interface system is the collection of architecture elements that sits between the source and target interfaces of an information flow and may transform the business data or call services. In many cases, the exchange of data is accomplished using a dedicated communication system, such as an EAI bus, which can be modeled by an information flow. The system as such is referred to as an interface system. Enterprise Application Integration platforms, message queues, request brokers, bus systems, or service orchestrators are typical examples of interface systems.

Both applications and components may model an interface system. An interface system may provide business support in its own right, in which case it should be modeled as an application. Or, it may provide technical support for the exchange of information, in which case it could be modeled as a component. The technical services called by an interface system can be specified for interface systems derived from a component or local component of the type **Service**. The interface systems specified for an information flow may be sequenced in order to specify the order that the services are called.

inventory

The inventory is the collection of all objects defined in Alfabet. The objects are often referred to as inventory objects. The inventory contains architecture elements that are actively used as well as any approved architecture elements planned to be used in the future.

inventory object

An inventory object is an object in the Alfabetinventory. Most inventory objects can be identified as architecture elements that may be affected by proposed architectural changes defined for a project'sto-be architecture.

Unlike an inventory object, a solution object is created in the context of an architecture solution and exists outside the Alfabetinventory. Once the architectural changes proposed by the solution are checked in to the inventory, the solution object becomes a real inventory object and will have an object state indicating that the object is planned. The object state must be manually changed to an active object state when operational use of the to-be architecture element begins, which is typically upon completion of the project's implementation.

issue 

An issue is a topic that needs to be addressed about a specified object in Alfabet. An issue type can be defined for the object as well as the architecture elements that may be targeted by the issue based on the issue type defined. For each issue, a priority can be defined as well as a status and planned completion date. Other issues that are relevant can also be associated with the issue. A demand may be derived from an issue. Issues may also be associated with features that shall be implemented to fulfill associated demands.

IT capability 

An IT capability is a basic capability such as an infrastructure service that is offered in the context of data center operations. Mainframe Operations, Database Management or Backup & Recovery are examples of typical IT capabilities.

IT capabilities can be associated with a technology that potentially provides support to the IT capability. The components that actually realize the technology for the IT capability can be further specified. For example, the components Oracle 10i and Oracle 11g enable the technology Oracle RDBMS, which realizes the IT capability Database Management. The use of a technology and its associated components at a specific location by an IT capability is defined in a technology support map

An IT capability can have an unlimited number of subordinate IT capabilities as well as an unlimited number of subordinate levels in the tree structure. However, an IT capability can be subordinate to only one IT capability at a given time.

IT landscape

The IT landscape describes the entirety of architecture elements and their relationships in the enterprise, including the application architecture, business architecture, information architecture, technical architecture, and deployment architecture.

IT strategy 

An IT strategy describes the target architecture that is best suited to support the enterprise's underlying business strategy. Because the IT strategy represents a long-term target for an enterprise, it is not defined for an explicit period of time. As such, the IT strategy provides direction for the master plan as well as architecture solutions that are developed in the context of projects. An IT strategy may be specified as a blueprint to be used as a guideline for IT strategy planning and master planning.

An IT strategy may have one or more IT strategy maps, which include the future strategic business supports specifically defined for the IT strategy. Because the IT strategy is subject to regular reviews and redefinition, multiple IT strategies may exist for an enterprise. For example, an IT strategy may be defined for an assumed market growth of 40 % as well as one for a market growth of only 20%.

IT strategy map

An IT strategy map is a constituent of an IT strategy that allows strategy planning to be divided into domains of responsibility, affinity, or commonality. An IT strategy may have multiple IT strategy maps. A different target architecture can be specified in the business support map defined for each IT strategy map. Each IT strategy map may feature a set of preferred ICT objects to be used as the provider of strategic business supports in the context of the business support map. In the case of non-IT support, a preferred set of organizations may be defined. IT strategy maps support the alignment of strategic business supports along business processes or domains in the context of the target architecture.

J

K

L

landscape

See IT landscape

layer

See platform layer

lifecycle

A lifecycle describes the succession of stages that an architecture element goes through. Many objects (for example, applications, components, standard platforms, business supports) in Alfabet have a lifecycle, although a lifecycle does not have to be defined for all objects. A lifecycle is comprised of lifecycle phases that describe the object's status of activity or production over time. Each lifecycle phase is aligned with its proceeding and succeeding lifecycle phase.

The lifecycle definition also includes the definition of the object's active period. The active period of an object is considered the period that the object is in production. Therefore, the active period of an object corresponds to the object's start and end dates. The active period may be configured to be aligned with one or more specified lifecycle phases that represent the period when the object is in production. Typically, the object state of the object will be specified as Active in the period between the start and end date.

The lifecycle definitions of dependent objects should be aligned with the lifecycle of the object that they have been defined for. For example, the lifecycles of any local components, information flows, and business

supports should be aligned with the lifecycle of the application that they are assigned to. In the case of components, the lifecycles of any local components, information flows, and technical services should be aligned with the component's lifecycle.

Furthermore, the lifecycle definition for an application may include the specification of predecessor and successor versions and the lifecycle definition of components may have successor versions. Successor versions will be assigned per default to the same ICT object as the original application/component, but this can be modified, as needed.

lifecycle phase

A lifecycle phase describes a status of activity or production in the lifecycle of an object. Typical lifecycle phases include, for example, Evaluation, Pilot, Production, Limited Production and Shut Down. The active period of the lifecycle is the period that the object is in production and is thus aligned with the object's start and end dates. One or more lifecycle phases may be left undefined. In this case, they are skipped. Each lifecycle phase is aligned with the proceeding and succeeding lifecycle phase that has been defined.

lifecycle status

See lifecycle phase.

local component 

A local component is a component defined in the scope of an application or standard component. Local components have the same attributes as standard components but are not reusable outside the context of the application or component they are defined for.

A local component inherits its authorized user and/or authorized user group from the application or component that provides it.

location 

A location is a geographic place that could be, for example, a country, city, building, or room. A device may be assigned to a location, which represents the physical location of the device. Locations can be hierarchically defined.

logical device

A logical device is a device that allows for the logical clustering of physical devices. A logical device can cluster a number of physical devices. A physical device can be clustered by a number of logical devices.

M

mandate

A mandate is a means to organize and structure the federated architecture of a holding company. The assignment of mandates to objects allows the holding company to structure the objects in the enterprise architecture in order to regulate visibility to objects across some or all subsidiaries. Only users explicitly assigned to a mandate will see objects with that mandate definition. An object that has not been assigned to a mandate is considered not to be owned by a mandate and is thus visible throughout the holding company to users with relevant access permissions.

The use of mandates in the Alfabet solution is optional. If mandates are implemented in an enterprise, the visibility of an object with a mandate assignment will take precedence over other concepts of access permissions in Alfabet. For example, an authorized user of an object must be assigned the relevant mandate to access the object that he/she is the owner of.

Within the context of mandates, the conventional rules governing access permissions apply. Thus, a user assigned to a mandate will only have Read/Write access permission to the objects made visible by the mandate if he/she has authorized access to the object as an authorized user, deputy, member of an authorized user group or discussion group or via rule-based access permissions, workflow contributor or assignee of an assignment.

mandatory property

Each object class in the class model has a preconfigured set of properties that serve to semantically describe the object class. Some standard properties are mandatory properties and are highlighted with a red star in editors.

No enforcement mechanism is implemented if a mandatory property is undefined unless the property is included in a standard or configured class key that enforces the unique definition of a new object. If the mandatory property is included in a class key, then an error message will be displayed if the property is not defined.

map view

A map view is a specific segment of a business support map that is relevant to the specific needs of a user. The map view is defined for a master plan map and contains a limited set of business processes/domains and organizations/market products in the business support map. Individual business supports that are not relevant for the map view can be hidden. An unlimited number of map views can be defined for a master plan map. In this way, a large and complex business support map can be scaled down to become manageable and relevant to a specific user.

market product

A market product is the physical artifact or service produced and marketed by the enterprise. A market product requires business support that is provided by applications. Object class stereotypes may be configured for the class **Market Product**.

market product group

A market product group is a container to logically structure market products in order to view, analyze, and communicate the company's product and service offering. There may be various ways to logically structure market products. Thus, any market product may be associated with multiple market product groups.

master plan

A master plan describes the medium-term to-be architecture required to fulfill the enterprise's long-term IT strategy. A master plan describes the entirety of business supports planned by the enterprise or an independent organizational entity of the enterprise. A master plan may have one or more master plan maps, which describes the future tactical business supports specifically defined for the master plan. A master plan may be specified as a blueprint to be used as a guideline for master planning.

master plan map

A master plan map is the constituent of a master plan that allows master planning to be divided into domains of responsibility, affinity, or commonality. A master plan may have multiple master plan maps.

A master plan map is a matrix made up of a set of organizations or market products and a set of business processes or domains and allows for the analysis of the alignment of planned business support. The master plan map may feature a set of preferred architectural elements to be used when defining tactical business support in the business support map.

A map view allows users to view only a section of the business support planned in a master plan map's business support map. Master plan maps may be organized into master plan folders.

master planning

Master planning is the process of defining master plans that align with the IT strategy defined in IT strategy maps. Master planning includes the definition of tactical business support in master plan maps. The tactical business supports defined in such maps build the middle-term planned support of business processes or domains that is provided by applications and ICT objects and is consumed in different organizations or by market products. Master planning also includes the process of analyzing and aligning the tactical and operational business supports regarding their timeframe, the business services provided as well as the business, technical alignment, and strategic alignment.

master platform 

A master platform is a generalization of a standard platform. A master platform is based on a platform template that structures platform layers and platform tiers. In contrast to a standard platform where the element is a component, a master platform element is a component category.

master platform category

A master platform category bundles and classifies content-specific master platforms. Master platform categories allow you to hierarchically structure the master platforms in your enterprise. Each master platform may be associated with only one master platform category.

master platform element 

A master platform element is the constituent of a master platform. A master platform element is associated with a component category.

measure

A measure is the instantiation of a measure type and allows rated and target measurements to be captured for the measure type for an active time series evaluation type. For example, a measure could provide the actually measured value 5 for the measure type Customer Satisfaction.

Rated measures can be assessed over time for the defined measure types for the enterprise in general. Target measures can also be assessed over time for objectives defined for a value node. Target and rated measures can be edited, as needed

measure type

Measure types describe a category of measurement that can be rated for the enterprise. A measure type could be, for example, Customer Satisfaction or Response Time. The instantiation of a measure type is a measure that represents the actual rated value of the measure type for an active time series evaluation type. Each measure type is assigned to a measure type category, which ensures responsibility and authorization for the actual measurement of the measure types in the measure type category.

Measure types may also be assigned to objectives defined for a value node. In this context, the value determined for such a measure type at a specific moment in time can be captured. Measure types can be rated in terms of an actual assessment of the objective and a target value can be defined for the measure types associated with the objective. Typically, a measure type is assigned to only one objective.

message flow

A message flow is a connector in a service diagram that indicates that information is transferred between the agents represented by swim lanes or between activities having the same agent. A message flow can connect an activity and event across different swim lanes (of different pools) or two activities in the same swim lane, in particular when a business object is transferred between two activities.

migration

A migration is the process in which applications, solution building block, and/or ICT objects are substituted by other applications/solution building blocks/ICT objects. In Alfabet, migrations are planned in the context of master planning or strategic planning. A migration typically includes migration rules, which provide an incremental description and target date of each required step of the planned transformation.

Migrations can be assigned to an IT strategy, master plan, and project. If the migration is assigned to an IT strategy or master plan, the prescribed business supports defined for the applications, solution building bocks, or ICT objects impacted by a migration rule can be checked for alignment with the planned migration.

A migration can be assigned to multiple migration groups.

migration group

A migration group is a container to logically structure migrations in order to view, analyze, and communicate the company's physical IT infrastructure. There may be various ways to logically structure migrations. Therefore, any migration may be associated with multiple migration groups.

migration rule

A migration rule is one step in the set of incremental steps required to realize a planned migration. A migration rule describes the transition from a predecessor object to a successor object in the step as well as the target date for the completion of the step. Predecessor and successor objects may be applications, solution building blocks, and/or ICT objects.

milestone

A milestone is a means to plan and track the progress and completion of projects or enterprise releases. Milestones can be defined in the context of a milestone template that bundles a set of predefined milestones, or they can be created as individual ad-hoc milestones independent of a milestone template. Adjustments to projected target dates and the reasons for date changes can be tracked and reconstructed in a milestone history. The realization of milestones is tracked on the level of the project group or enterprise release.

milestone template

Milestone templates are defined according to the project methodology pursued by the enterprise in order to assess the stage gates or milestones of the project methodology or enterprise release cycle.

A milestone template bundles a set of milestones including the definition of the color scheme used for tracking and reporting milestones and the sequence and tolerance period for each milestone. Users may either assign a configured milestone template directly to a project or enterprise release or, in the case of projects, they may select a configured project template and assign it to the project.

Projected target dates are automatically set for each milestone based on the configuration defined in the milestone template, but these dates may be manually edited for each milestone. Users may delete any milestone from a project or enterprise release that they deem unnecessary. Only one milestone template may be assigned to a project or enterprise release. The realization of milestones can be tracked for the enterprise release or, in the case of project milestones, on the level of the project, project group, or bucket.

monitor 

Alfabet provides a variety of monitors to alert users about changes that have occurred to specific objects and may require further activity, reviews that did not occur as intended, or transactions that were planned to occur according to the documented plans. Such monitors alert the subscribed users through e-mail notifications. Monitoring is performed in regular intervals for a specific period of time. Alfabet currently provides activity monitors, date monitors, inactivity monitors, notification monitors, and consistency monitors.

N

navigation page

Navigation pages allow an enterprise to configure a pages with links and information for the enterprise's users in order to efficiently guide them to specific functionalities in the Alfabet interface. The navigation page that serves as the start page is assigned to a user profile. Navigations page can be configured to provide a variety of different kinds of support to users in their work in Alfabet. A navigation page can be freely configured and could include, for example, links to specific functionalities in Alfabet, links to other navigation pages, informational text or images about enterprise-specific workflows in the Alfabet solution, standard framework approaches like Zachmann, or URL links to internal documents or the Web.

network 

A network is a system of devices and other physical networks that are connected via network routes. A network may be either a source or destination of a network route. The definition of the network includes its spatial scope such as WLAN, LAN, Internet, etc. A network may have subordinate networks.

network route

A network route represents the routing of a network or device to another network. Network routes are based on a device of the type Router and provide the physical infrastructure necessary to enable information flows deployed via a deployment.

notification monitor

A notification monitor is a type of monitor that allows email notifications to be automatically triggered based on configured Alfabet queries or native SQL queries. The queries specify the targeted objects and their object class properties as well as the users who shall be notified about the objects found by the queries.

O

object

An object is a constituent of the Alfabet architecture that generally describes an element in the enterprise architecture. An object belongs to an object class defined for the Alfabet meta-model. Specific objects may also be referred to as architecture elements, artifacts, or instances. Many objects in Alfabet have a start and end date that specifies the planned period of activity for the object, a lifecycle definition, and an object state that distinguishes an object as actively used, planned to be used, or to have been used in the past.

object class

An object class is a construct that semantically describes the objects that are instantiated for that class. Object classes in the class model have a specified set of standard object class properties and may also have custom properties that have been configured by a solution designer. Object classes typically reference other object classes, thus creating a semantic network that can, for example, describe the enterprise's IT landscape.

object class property

Each object class in the class model has a preconfigured set of properties that serve to semantically describe the object class. Some standard properties are mandatory properties and are highlighted yellow in Alfabet editors.

object class stereotype

An object class stereotype is a sub-classification of an object class. A permissible object class can have multiple object class stereotypes, each of which captures a specified set of attributes, reference data, and

class configurations. For example, the object class Application that may have the object class stereotypes Business Applications and Technical Applications.

Only a limited number of object classes support the configuration of object class stereotypes.

object cockpit

An object cockpit provides users with an immediate and transparent overview of data for a selected object. It is as an abbreviated and focused presentation of object data, typically from a particular perspective such as an architecture perspective or a strategy perspective.

Multiple object cockpits can be available per object profile. For example, one object cockpit for the object class Application might display data relevant for understanding the application in the as-is architecture, another cockpit might display data relevant to master planning issues, and a third cockpit might display relevant data for cost and budget issues. The object cockpit is a configuration that is available in addition to the more comprehensive object profile.

object profile

The object profile summarizes information that is relevant to a selected object. The object profile typically displays object class properties and their values for the object as well as workspaces with views and reports relevant to the selected object. Users can open the available views and define or edit the data about the selected object.

Typically, object profiles are configured specifically for the needs of the user community. Thus, various object profiles with different data and functionality may be available for the same object class, depending on the user profile with which users access Alfabet. Each object profile may have multiple object cockpits associated with it that have been configured to visualize a specific set of data about the object.

object state

An object state describes the operational status of an object in the enterprise. The object state indicates whether an object is actively used, planned to be used, or has been used in the past. Because an object's start and end dates specify the planned period of activity for the object, the object state should be changed from **Plan** to **Active** once the object's start date is reached. Equally, the object state should be changed from **Active** to **Retired** when the object's end date is reached. If the lifecycle concept is available for an object class the object's start and end dates will initially be aligned with the active period of the object.

An object's object state can be changed by a user with Read/Write access permissions. For objects like applications or components, relevant object dependencies must be taken into consideration when the object state is changed. For applications, for example, the user changing the object state should consider whether the object state should be propagated to the application's associated information flows or business supports.

objective

An objective expresses a refinement and specification of a value node that can be acted upon. Objectives may only be assigned to one value node and cannot be shared across such value nodes. A value node may have multiple objectives defined. An objective inherits the user authorization definition of the value node that it is assigned to.

Every objective should be associated with at least one measure type in order to define target measures for the objective. This allows for the validation of the realization or attainment of strategic intentions to be measured and tracked in terms of actual ratings vs. targets.

operational aspect

An operational aspect is a facet of the business that may be supported by the IT. Standard classes that represent operational aspects in Alfabet include Brand, Customer Segment, Market, and Sales Channel although other object classes or object class stereotypes may be implemented as operational aspects.

Multiple operational aspects can be created for each configured aspect class to express the individuals facets of business that require support. For example, aspects for the class Market could be Capital Markets, Retail Banking, Emerging Markets, and Direct Banking.

Operational aspects are relevant in the context of business support planning and analysis and allow the planner to understand the aspects of the business that an organization, market product, business process, or domain contribute to. Understanding the contribution of individual business supports to operational aspects of the business sheds light on where future support is required in order to realize the enterprise's IT strategy. Operational aspects can be defined for operational business supports, tactical business supports, strategic business supports, and solution business supports. Potential gaps or redundancy in the support of operational aspects can be understood and analyzed in the context of business support maps.

Please note that the concept of operational aspects has no association with the concept of aspect evaluations, which are implemented to assess the performance of applications and components.

operational business support

An operational business support is an active or planned business support that is currently provided or will be provided as a result of ongoing development or roll-out activity.

An operational business support can be provided by an application or, in the case of support via internal or external services, by an organization. The operational business support provides support to either a business process (and business process version) or domain and can be provided to either an organization or market product.

organization 

An organization describes an administrative or functional unit in the enterprise. Organizations form a self-referential hierarchy. An organization may have an unlimited number of subordinate organizations but only one ascendant organization. An organization may have primary responsibility for an object in the same way that an authorized user has or it may by defined as having an important role in relation to an object.

Organizations are supported in their business activities through the business support and business services provided by applications.

Object class stereotypes may be configured for the class **Organization**.

organization group

An organization group is a container to logically structure organizations. An organization may be associated with multiple organizational groups.

owner

See authorized user

P

peripheral

A peripheral is an element in the IT landscape that is outside of the core scope of the Alfabet solution. A peripheral typically represents an application that is managed by business partners, such as an EDI gateway or a B2B marketplace.

Peripherals are connected to applications in the IT landscape by means of information flows. In contrast to an application, a peripheral's local components or business services as well as the details of its technical platform are of no relevance in the Alfabet solution.

person

See user

personal item

A personal item is any object that the user owns. The user is thus defined as the object's authorized user.

perspective

A perspective is a specific view that can be applied to a value node. Typical perspectives include Innovation, Process, Customer and Finance. Perspectives are typically used to verify completeness of the strategy

definition for an enterprise or sub-organization. Perspectives are also used to determine lineage between strategies as expressed by value nodes. A perspective can be assigned to multiple value nodes.

When the value nodes owned by an organization are analyzed, the rated and target measures defined for the objectives associated with the value node can be understood in terms of the associated perspective. In this way, a balanced scorecard analysis is supported that allows the realization of strategic intentions to be understood based on perspectives critical to the enterprise.

physical device

A physical device is a device that represents a piece of hardware that software runs on. A physical device can be associated to a logical device by means of clustering. Clustering features a many to many relationship between physical and logical clusters.

platform

A platform describes the technical architecture in terms of the assembly of local components that a specific application or component runs on. This assembly is based on a platform template organized in a matrix-like structure with platform layers and platform tiers defining the axes. The structure reflects the partitioning of the platform in terms of distribution across multiple hardware and software groups.

platform element 

A platform element is a constituent of a platform. Platform elements are associated with standard components or local components. Any qualitative aspect of local components as well as information about business services, information flows, or business data is disregarded in the context of a platform.

platform layer 

A platform layer is a dimension of the partitioning of a platform. Platform layers represent a grouping of components based on the level of abstraction of services that components within a layer provide.

Usually, a platform layer depends on the layers below it. A typical layered architecture would comprise a business layer, where business logic resides, a software infrastructure layer, where operating system and infrastructure services reside, and a hardware layer. Platform layers are organized with platform tiers in a platform template.

platform template 

A platform template defines a standard set of platform layers and platform tiers and thus defines a reference grid for placing components in standard platforms and platforms.

A platform template structures the platform in two dimensions. The horizontal dimension represents the platform tiers, a grouping of components within a platform according to the functionalities deployed on separate physical computers. The vertical dimension represents the platform layers, a grouping of components that classifies the services provided within a layer.

The platform template can also display a structured description of the application architecture. For example, this could include:

- The logical layer of the architecture that describes components of applications/components on a logical or business level.
- The technical layer that describes the implementation or realization of the application/component.
- The physical layer that describes the runtime environment and allocation of the applications/components in networks and on different devices.

platform tier 

A platform tier is a dimension of the partitioning of a platform. Platform tiers represent a grouping of components in terms of functionality that can or must be deployed on separate physical machines in order to accomplish better modularization and scalability.

Usually, each platform tier communicates only with the tiers next to it. Typically, a four-tier architecture would comprise a database server tier, an application server tier, a Web server tier, and a client tier. Platform tiers are organized with layers in a platform template.

policy 

A policy is a guideline or rule of business or conduct that must or should be followed by the enterprise and its constituents. A policy can be associated with a value node or principle. The policy will have a start and end date and can have a review date specified.

pool

A pool is a boundary element in a service diagram that serves as a framework for the service diagram elements that visualize a single business process. A pool can consist of a number of swim lanes that represent the different agents conducting the business process visualized by the service diagram. The service diagram elements that represent internal activities are placed within the boundaries of a pool. External activities or events, like a start or end event, are typically placed outside the boundaries of the pool.

portfolio

A portfolio is a bubble chart that represents the relative performance of a set of objects that belong to the same object class in two or three independent dimensions of measurement. Typical portfolios include a group of applications that are owned by a specific organization or used by a particular process.

The portfolio consists of an X-, Y-, and if necessary, power-axis. The power-axis is the Z-dimension and is reflected via bubble size. Either an indicator type, evaluation type, or prioritization scheme can be assigned to a portfolio axis. In Alfabet, a portfolio may also be displayed as a BCG quadrant. Access to portfolios can be controlled via user profiles. A user can view any portfolio that has either no user profile defined or the same user profile as that which he/she is currently logged in with.

preview

A preview is a shortened version of the object profile view that is displayed below some tabular reports and summarizes the most important aspects of an object.

primary language

The primary language is the language of the base culture specified for your enterprise. The primary language is the default language and will be displayed when users open the Alfabet user interface upon first login. Users can change teh rendering of the user interface from the primary language to a secondary language as needed.

primary object

A primary object is an architecture element that is assigned to a domain, whereby the domain owns the object and defines the principal functional context of the object. In this case, the domain is referred to as the primary domain for the object. Architecture elements that can be assigned to a domain include business functions, business objects, business processes, applications, ICT objects, components, standard platforms, and vendor products.

A domain can have an unlimited number of primary objects assigned to it, but an object can have only one primary domain. An effective service-oriented architecture requires disjoint sets of primary domains in the enterprise architecture.

prioritization scheme

A prioritization scheme is a weighted composite of a set of evaluation types. A prioritization scheme is typically used to determine a prioritized ranking for a set of objects, for instance during budgeting and prioritization of projects or to define the axis of a portfolio.

Prioritization schemes aim at aggregating evaluations along complementary evaluation dimensions. Whereas evaluation types are aggregates of indicators for semantically-related performance measurements of objects, prioritization schemes are typically aggregates of semantically-unrelated (or loosely related) performance measurements for objects. However, prioritization schemes are assigned to a portfolio usually describe performance along dependent dimensions.

principle

A principle is an overarching guideline that directs the enterprise architecture definition and decision-making across the company. An organization is responsible for the principle. A principle may have an unlimited number of subordinate principles but only one ascendant principle.

profile

See object profile or user profile.

project

A project is an activity undertaken in order to achieve a specified goal in the IT landscape. Typically, projects are based on configured project stereotypes. A project may have unlimited sub-projects (projects of the subordinate project stereotype) but only one ascendant project (a project of the ascendant project stereotype). Projects may be grouped into project groups as well as assigned to buckets for budgeting and prioritization purposes.

Depending on the configuration defined for a project stereotype, it may be possible to bundle demands to projects, document the as-is architecture that may be impacted by the project, plan the to-be architecture for the IT landscape, plan and assess the project's costs in a business case, capture the skill requests and resource requests needed for the project and manage the project resource plan, and monitor project target dates via milestones.

Furthermore, you can also work with project scenarios based on an existing project in order to plan alternative what-if scenarios in terms of costs, resources, scheduling, and to-be architectures. Finally, project baselines may be created for a project in order to understand the scope of changes that have occurred in a project.

project baseline

A project baseline captures the scope of its base project at the time that the project baseline is created. The project baseline provides a ReadOnly view of the base project's data and allows the deviation of the base project from its original scope to be tracked over time in terms of the total project costs and income, skill request and resource request costs, evaluations, scheduling and achievement of project milestones, and the changes made to the to-be architecture.

project evaluation type

A project evaluation type is an evaluation type that has been configured to allow project monitoring to occur for projects. The projects can be monitored regarding the achievement of target values for the indicator types assigned to the project evaluation type.

project group 

A project group is a container to logically structure projects. There may be various ways to logically structure projects. Therefore, any project may be associated with multiple project groups. Project groups support lump sum budgeting and prioritization of the projects contained in the group.

project resource plan

A project resource plan is the detailed planning and scheduling of the skill requests and resource requests for a project. In contrast to high-level project planning of project resources, which focuses on the general scope of skill requests and resource requests needed for the project, the project resource plan targets the management of resources including the persons and organizations responsible for carrying out the associated tasks as well as the overall costs and scheduling of the required skills and resources.

project solution

A project solution allows alternative projects to be drafted and considered. Multiple project solutions may be defined for a specific project, but only one project solution can be selected as the approved project solution. Once a project solution is selected, it is considered a project in its own right and replaces the original project that it was defined for in the project hierarchy. Project scenarios are an extension of the project solutions concept but provide much more flexibility and functionality in the planning and implementation of what-if scenarios.

project scenario

A project scenario is an alternative what-if scenario to be explored, planned, and potentially implemented for the project. The project scenario may target alternative to-be architectures, project costs and income, the scope of skill request and resource request required for the project, various evaluations of different aspects of the project, and different versions of project dependencies and time schedules.

A project scenario is an independent copy of the project and includes its start date and end date, custom properties as well as the base project's business caseas-is architecture, to-be architecture, requested skills and resources, value nodes, contract deliverables, and input demands. All subordinate projects will also be copied.

Each project scenario can be further defined and modified without impacting the base project. ´The project scenario can be merged at any time to the base project in order to update the base project. The project scenario as well as the base project continue to exist and both can be further modified, if needed.

project stereotype

A project stereotype represents a level in the project management framework. The project hierarchy is made up of an arbitrary number of object class stereotypes defined for the class `Project`. An unlimited number of objects can be created in Alfabet for each project stereotype.

The project hierarchy can be configured as a linear hierarchy or as a branched tree structure that allows complex project structures and hierarchies of stereotypes to be captured. Each project stereotype may define multiple subordinate project stereotypes. The top-level project stereotype represents the most abstract level of a project and the lowest level project stereotype represents the most granular level of project tasks that must be acted upon in order to realize the project. For example, a typical linear project might be made up of the following project stereotypes: Level 1: Program, Level 2: Statement of Work, Level 3: Project, Level 4: Sub-Project, Level 5: Project Activity.

The name, number, and hierarchical ordering of the project stereotypes are configurable. A project stereotype is similar to an object class. Like an object class, the projects based on a project stereotype are instances of the project stereotype. For example, the project stereotype Project Activity will have objects called project activity. Each project stereotype may have only one ascendant project stereotype and only one subordinate project stereotype. However, projects of a specific stereotype do not have to be to be associated with a project of the ascendant project stereotype nor with projects of the subordinate project stereotype.

The definitions that can be made for each project stereotype must be configured. Therefore the views that are available may differ for each project stereotype in the hierarchy. For example, the configuration will determine for which project stereotype users can bundle demands to projects, document the as-is architecture, plan a to-be architecture, assess alternative project scenarios, track costs via cost centers, and monitor projects via milestones.

project template

A project template consists of a configured milestone template, which bundles a set of milestones, and a skill request template, which groups a set of skills. A project template can be assigned to a project and its sub-projects in order to ensure that the project hierarchy has a consistent definition of milestones and skill requests.

proposed information flow

A proposed information flow is typically a imported via a discovery run and serves as a potential information flow for a source or target application or component platform. A proposed information flow can explicitly be changed to a "real" information flow. In this case, the information flow will inherit all attributes defined for the proposed information flow and the proposed information flow will be deleted from the Alfabet database.

proposed local component

A proposed local component is typically a discovered component that has been imported via a discovery run and serves as a potential technology to be included in an application or component platform. Proposed local components are usually used to upgrade to simple technology changes that don't require the planning and investment needed for significant architectural changes, which would typically be scoped in a project solution. Proposed local components can also be used to replace existing local components for a simplified versioning of technology. A proposed local component can explicitly be changed to a "real" local component. In this case, the local component will inherit all attributes defined for the proposed local component and the proposed local component will be deleted from the Alfabet database.

proxy

A proxy is a person who is assigned as an alternate user to be responsible for assignments or workflow steps in case of the planned absence of the original responsible user. The proxy can be assigned individual assignments and workflow steps as well as all assignments and workflow steps. The proxy user will have the necessary access permissions to the assignments or workflow steps that he/she is responsible for via the proxy assignment.

Q

query

See alfabet query.

R

reference model

A reference model is a hierarchically structured collection of framework groups and serves as a means to define a semantic structure for objects managed in Alfabet regardless of the object class that the object pertains to. Users can navigate through a reference model in Alfabet.

release

See stack

release status

A release status describes the state of approval for or agreement about an object in the enterprise. Typical status values could be, for example, `Draft`, `Described`, `Reviewed`, `Approved`, `Rejected`, and `Retired`. However the release statuses will largely depend on the object class that they describe. For example, the release statuses for an architecture element such as an application, component, or standard platform is typically used to express agreement to the state of the documented information whereas, the release statuses for a planning artifact like a demand or project reflects an approval status. The release status for an assignment describes the state of progress or completion of the task that the assignment is about.

An object's release status can be changed by a user with Read/Write access permissions. For all release statuses that are configured as non-editable release statuses, users will not be able to edit the attributes in

the object's editor and page views. Users will see the  icon at the top of the object profile of an object that may not be edited due to its current release status definition.

report

A report is a configured display of data from the Alfabet database. Alfabet provides a number of standard reports, however customized reports can also be configured by a solution designer.

resource

A resource is any entity that is needed as a source of support for another entity in the enterprise. A resource may be provided by many different object classes such as an organization, application, device, deployment, etc. A resource is typically requested by a project, service product, or organization.

resource request

A resource request is typically made by a project, application, component, device, deployment, service product, standard platform, or organization in order to fulfill the need for a resource. Object class stereotypes are typically configured for resource requests in order to distinguish the different types of resources based on the different types of providers which could be an application, component, device, deployment, service product, standard platform, or organization. A resource is associated with a cost type and includes the specification of the amount needed for the resource as well as the total cost of the requested resource.

responsibility

A responsibility defines the functional relationship that a specific user or user group has for an object. The responsibility can be specified by defining a role for the selected object.

Unlike an authorized user, a user or organization with a role is not required to maintain an object's data. Roles are defined for informational purposes only and provide detail about users or organizations that may have information about or a stake in the object. The definition of a role for an object does not impact access rights.

responsible user

See authorized user or role.

risk

A risk refers to the likelihood that a particular object in the enterprise's IT landscape is exposed to a given threat and potentially targeted by a given attack. The definition of a risk includes the assessment of the

potential damage caused by the risk as well as the potential probability that the risk could occur. A risk mitigation may be defined that describes how the risk might be reduced or avoided.

A risk can be defined explicitly for a specific risk object or it can be added to the risk object by means of a configured risk template grouping a standard set of risks and, if relevant, their suggested mitigations.

risk management group

A risk management group is a container to logically structure risk objects in order to view, analyze, and communicate risks to the company. Each object assigned to a risk management group is considered a risk object. It is the risk object that is evaluated and assessed in the context of the Risk Management capabilities. This ensures that this information is only visible to users in the user community that have access to the risk object. An object may only be assigned once to a selected risk management group, but it may be associated with multiple risk management groups, allowing the same risk object to be considered in different contexts.

risk management template

The risk assessment in Alfabet consists of two phases. A first phase is the **Risk Assessment** phase and allows the enterprise to assess which objects are potentially at risk. This phase allows objects to be assessed in order to determine whether they should be included in the risk evaluation. The second phase is the **Risk Evaluation** phase and allows the enterprise to document and evaluate the risk to the objects targeted by the risk evaluation. This includes the potential cost of damage to the object and that probability of damage to the object as well as the means to mitigate the risks. This approach allows the enterprise to streamline the risk evaluation process and specifically target the objects that are considered most at risk.

risk mitigation

The risk mitigation articulates a step to avoid, reduce or contain the risk. The risk mitigation may be based on a predefined risk mitigation template. For each risk mitigation, the architecture elements that might be impacted by the risk mitigation can be documented. Furthermore, a demand can be created to express the need to address the risk mitigation in the IT architecture. Once the demand to address the risk mitigation has been articulated, a project can be created to implement the risk mitigation.

risk mitigation template

A risk mitigation template captures a preconfigured definition of a risk mitigation including the name of the risk mitigation, the target date when the risk mitigation should be implemented, and a number used to prioritize the risk mitigation. A risk mitigation template can be created for a specific threat, threat group, or risk mitigation template category. Each risk mitigation template can be assigned to one risk mitigation template category.

risk mitigation template category

A risk mitigation template category bundles and classifies content-specific risk mitigation templates. Risk mitigation template categories allow you to hierarchically structure the risk mitigation templates in your enterprise. Each risk mitigation may be associated with only one risk mitigation template category.

risk object 

A risk object references an object that has been determined to exceed a specified base risk exposure and must therefore be assessed for risk damage and risk probability in the context of the risk management capability.

For each risk object, values must be provided as answers to the questions posed in a configured risk questionnaire. Additionally, the potential risks can be specified including their potential damage, their probability of occurrence, and a mitigation to reduce or avoid the risk.

Risk objects can be structured in a risk management group in order to allow their risk to be understood and analyzed from different perspectives. A risk object can only be assigned to one risk management group. However, an object (for example, an application) may be referenced by two different risk objects belonging to different risk management groups, thus allowing for the analysis of potential risks from different perspectives.

risk template

A risk template bundles a set of risks and, if necessary, their suggested mitigations. Multiple risk templates can be assigned to a class-based risk management template.

When a risk object is assessed, the risk template can be selected in order to add a standard set of risks and, if relevant, their mitigations to the risk object. Irrelevant risks can be removed from the risk object and additional risks that may not have been captured in the risk template can be defined for the risk object.

role

A role defines the functional relationship or responsibility that a user or organization has to an object. A role is based upon a configured role type that is configured for an object class. Roles are defined for informational purposes only and provide detail about users or organizations that may have information about or a stake in the object. The definition of a role for an object does not impact access permissions.

role type

A role type is configured in order to allow a functional role to be defined for objects in a specified object class. Role types can be configured to be explicitly available for a user or organization or both.

root object

A root object is an object defined for the highest level of a hierarchical explorer tree structure. Root objects can be found in the hierarchies associated with organizations, business process models, business functions, or application groups.

rule 

A rule is a condition that describes how business processes are connected to one another. Rules are displayed in business process diagrams. Available rule types include AND, OR, XOR, Synchronisator, Connector.

rule-based access permission

A rule-based access permission is a rule based on an alfabet query or native SQL query that specifies access permissions for a found set of objects or object class.

S

secondary language

A secondary language is any language that is not the primary language and that has been configured for the enterprise. The user interface may be rendered in and object data may be captured in a secondary language. Multiple secondary languages may be configured for the Alfabet solution.

sequence flow

A sequence flow is a connector in a service diagram that connects an activity and event within the same swim lane. A sequence flow indicates the order in which the activities are conducted. In other words, an activity is connected with its predecessor activity and its successor activity by means of sequence flows.

service contract

A service contract captures the fees that an organization pays in order to provide services. The service fees constitute the costs to use the applications and ICT objects needed to provide the services. A service contract is based on a service contract templates.

service contract template

A service contract template defines the basis for pricing for service contracts. The price for the service charges of a service contract is calculated as follows: OneTimeFee*FirstTimeUsage + Base Price + Usage-Price*Usage. The values for **One Time Fee**, **Base Price** and **Usage Price** are predefined in the service contract template that the service contract is based on.

service diagram

A service diagram is a diagram associated with a business process that visualizes the dependencies between its sub-processes (activities) or business services requested by the business process. A business process can have multiple service diagrams. However, a service diagram belongs to only one business process.

A service diagram may contain diagram elements of the types event or activity, simple connectors of the types sequence flow, message flow, or association, complex connectors of the type gateway as well as boundary elements of the type pool or swim lane. In addition, business objects, business services, and business functions can also be visualized in a service diagram. Additionally, business services and business functions can be connected to other service diagram elements by associations.

A service diagram can reference another service diagram by a link associated with an activity. Service diagrams can only be viewed at the level of business processes.

The design of service diagrams in Alfabet follows BPMN conventions.

service level agreement (SLA)

An SLA is a service level agreement that is defined for a service product. The SLA captures information such as a volume base that is relevant for the service product (for example, Service Desk Support Hours, Target Resolution Time, Defect Backlog Size Limit, Maximum Number of Test Failures, etc). Only one SLA may be assigned to a service product. The service level agreement automatically inherits the authorized user definition of the service product that it is assigned to.

Object class stereotypes may be configured for the object class **Service Level Agreement**. This allows for different attributes to be used for different types of service level agreements.

service product

A service product is a service that is owned by an organization and made available to other entities. A service product consists of one or more service product items, which constitute the service product. Each service product item is associated with an object such as an application, component, ICT object, deployment, device, standard platform, organization, or other service product, which provides the service product item to the service product.

Each service product may have one SLA (service level agreement) assigned to it.

Service products are organized according to the organization that owns them. Each organization and the service products that it owns constitute a service product catalog. Additionally, service products may be

structured in one or more service product groups independent of the organization that provides the service products.

A service product may be defined as a contract deliverable for a contract, whereby the service product is consumed by the contract. A service product may be assigned to a market product in order to indicate that the service product is made available via the market product.

service product item

A service product item is an entity that supports a service product. The service product may be constituted by one or more service product items. Each service product item is associated with an object from a permissible object class (such as an application, component, ICT object, deployment, device, standard platform, organization, or other service product) that provides the service product item to the service product.

A name may be defined for the service product item to capture the purpose of the object associated with the service product item. For example, the service product User Support Service might consist of the service product items Application User Support Services which is provided by applications, Deployment User Support Services which is provided by deployments, and Device User Support Services which is provided by devices.

The service product item may also define a volume to allow representation of a bill of service (for example, if a service product requires two instances of a specific server because the service product must provide for high availability).

A service product item inherits the authorized user and/or authorized user groups from the service product that it is assigned to.

Object class stereotypes may be configured for the class **Service Product Item**.

skill

A skill is a field of expertise necessary to fulfill a specific aspect of a project. The skill can be assigned to a skill request template that specifies the head count or man days and the expected cost required by the skill in the context of a particular kind of project. Users making a skill request thus have predefined information that helps to estimate the resources and costs involved if the skill is implemented in the project.

Not all skills required to realize a project need to be estimated. Rather, the skills that the enterprise deems scarce should be projected when preparing for the submission of project for prioritization and budgeting.

skill offer

A skill offer represents a skill that can be provided by an organization. The organization has been specified to fulfill a skill request for a project.

skill request

A skill request defines a request by a project for a required skill necessary to fulfill a specific aspect of the project. A skill request allows you to define the type of skill required as well as the head count or man days required for the project, the organizations that will provide the skill, and the expected cost required by the skill. Skill requests can be bundled in a skill request template and assigned to a project template, thus streamlining the process of defining the skills required by a project.

skill request template

A skill request template allows skills to be specified in order to plan resources for a project. A skill request template specifies the head count or man days required for the skill and the expected cost required by the skill in the context of a particular kind of project. Users making a skill request thus have predefined information that helps to estimate the resources and costs involved if the skill is implemented in the project. Skill request templates can be assigned to a project template in order to streamline the process of project planning.

solution application

A solution application is a proposed new application or proposed new version of an application specified in the to-be architecture defined for a project. Object class stereotypes may be configured for the class **Solution Application**. These object class stereotypes will typically be aligned with the object class stereotypes available for the class **Application**.

solution building block

A solution building block supports functional planning as endorsed by TOGAF. The notion of architecture building blocks (ABBs) can be captured in Alfabet via domains. By configuring a domain stereotype on the lowest level of the domain hierarchy to capture ABBs, the functionalities that the enterprise realizes can be specified via an ABB's association with business functions.

solution business process 

A solution business process is a business process proposed in the context of a solution business process model. The solution business process is a copy of a real business process that exists in the inventory and can thus be modified without changing the actual business process in the inventory. Only after the solution business process model is checked in to the inventory will the solution business process overwrite the existing business process.

Within the context of a solution business process model, business service requests, domains, and organizations can be defined for the subordinate solution business processes. These modifications are made within the solution business process model and do not affect the inventory objects they are based on.

Once a solution business process model is approved, the solution business process model and all solution business processes can be checked in to the inventory and thus become inventory objects.

solution business process model 

A solution domain is a proposed alternative for a domain hierarchy. A solution domain hierarchy is made up of solution domains. Multiple solution domain models may be defined, but only one of the solution domain models may be approved for check-in. Once a solution domain model is checked in to the inventory, it overwrites the existing domain model.

As long as a solution domain model does not have an approved status and has not been checked-in to the inventory, the solution domain model and its solution domains are considered to be "shadow" objects — copies of the domain model and domains that exist only in the solution and do not appear in the inventory. Any modifications made within the solution domain model do not affect the inventory objects they are based on as long as the solution domain model has not been checked-in to the inventory.

solution business support

A solution business support is a proposed business support specified in the context of the to-be architecture for a project. A solution business support may be based on a solution application or an existing application. A solution business support describes the proposed IT support for an organization/market product in supporting a business process/domain. A solution business support can also be defined to plan the retirement of an operational business support.

solution component

A solution component is a proposed new component or version of a component specified in the to-be architecture defined for a project. Object class stereotypes may be configured for the class **Solution Component**. These object class stereotypes will typically be aligned with the object class stereotypes available for the class **Component**.

solution device

A solution device is a proposed device defined for an ICT object that is specified in the to-be architecture defined for a project. As a solution object, the solution device exists outside the inventory until the architectural changes proposed by the solution are checked in to the inventory. After check-in, the solution device becomes a real inventory object and will have the object state**Plan**. The object state must be manually changed to **Active** when operation begins.

solution domain 

A solution domain is a domain proposed in the context of a solution domain project. The solution domain is a copy of a real domain existing in the inventory and can thus be modified without changing the actual domain in the inventory. Only after the solution domain project is checked in to the inventory will the solution domain project overwrite the existing domain model.

Within the context of a solution domain project, subordinate domains can be added or removed. Any business functions, business objects, business processes, applications, ICT objects, components, standard

platforms, and vendor products that are assigned to a deleted solution domain can be assigned as a primary object or associated object to a successor solution domain. These modifications are made within the solution domain project and do not affect the inventory objects they are based on.

Once a solution domain project is approved, the solution domain project and all solution domains can be checked in to the inventory. The new solution domain thus overwrites the original domain in the inventory. Any new objects and references created for the solution domain project are also written to the inventory.

solution domain project

A solution domain project is a proposed alternative to a section of the current domain model. The domain that is the parent domain to all domains targeted for change is checked out of the inventory, thereby creating a solution domain project. The targeted domain and its sub-domains are added to the solution domain project as solution domains. Any domain in the domain model may only be the target of a single solution domain project at a given time. An error message will be displayed if a user attempts to check out a domain that is already involved in another solution domain project.

The solution domain project and its solution domains are considered to be "shadow" objects — copies of the original domain and its sub-domains that exist only in the solution domain project and do not appear in the inventory. Once the domain has been checked out and the solution domain project created, any further changes made to the inventory domain will not be written to the corresponding solution domain.

Within the context of the solution domain project, the solution domain hierarchy can be restructured by creating and deleting solution domains. Any objects referencing a primary domain or associated domain that is proposed for deletion can be reassigned to a successor domain.

Once a solution domain project is assigned the approved release status, it can be checked back in to the inventory. The target domain and its sub-tree in the inventory will be deleted in the domain model and replaced by the new domain hierarchy that was created in the solution domain project.

solution information flow

A solution information flow is a proposed information flow specified for a solution application or solution component specified in the to-be architecture defined for a project. As a solution object, the solution information flow exists outside the inventory until the architectural changes proposed for the to-be architecture are checked in to the inventory. After check-in, the solution information flow becomes a real inventory object and will have the object state **Plan**. The object state must be manually changed to **Active** when operation begins.

solution local component

A solution local component is a proposed local component defined in the scope of asolution application or standard solution component specified in the to-be architecture defined for a project. As a solution object, the solution local component exists outside the inventory until the architectural changes proposed for the to-be architecture are checked in to the inventory. After check-in, the solution local component becomes a real inventory object and will have the object state **Plan**. The object state must be manually changed to **Active** when operation begins.

solution map

A solution map allows the solution business support to be planned for the to-be architecture defined for a project. Multiple solution maps can be created for a project. The solution map displays a matrix with a set of organizations (or market products) on one axis and a set of business processes (or domains) on the other axis. A solution business support can be defined in the solution map for any solution applications created for the project's to-be architecture as well as any applications defined in the project's as-is architecture.

solution object

A solution object is any architecture element that is proposed in the context of a to-be architecture for a project. Solution objects may be based on objects assigned to the project's as-is architecture or may be created from scratch in the context of the to-be architecture. Solution objects include solution applications, solution components, solution devices, solution information flows, solution local components, solution peripherals, solution standard platforms, solution technical services, and solution business supports. A solution object exists outside the inventory until the architectural changes proposed in the to-be architecture are checked in to the inventory. After check-in, the solution object becomes a real inventory object and will have the object state **Plan**. The object state must be manually changed to **Active** when operation begins.

solution peripheral

A solution peripheral is a proposed peripheral that is connected to a solution application by means of a solution information flow specified in the to-be architecture defined for a project. As a solution object, the solution peripheral exists outside the inventory until the architectural changes proposed for the to-be architecture are checked in to the inventory. After check-in, the solution peripheral becomes a real inventory object and will have the object state **Plan**. The object state must be manually changed to **Active** when operation begins.

solution standard platform

A solution standard platform is a proposed standard platform defined for a solution application or solution component that is specified in the to-be architecture defined for a project. As a solution object, the solution standard platform exists outside the inventory until the architectural changes proposed for the to-be architecture are checked in to the inventory. After check-in, the solution standard platform becomes a real inventory object and will have the object state **Plan**. The object state must be manually changed to **Active** when operation begins.

solution technical service

A solution technical service is a proposed technical service created for a solution component of the type **Service** that is specified in the to-be architecture defined for a project. As a solution object, the solution technical service exists outside the inventory until the architectural changes proposed for the to-be architecture are checked in to the inventory. After check-in, the solution technical service becomes a real

inventory object and will have the object state **Plan**. The object state must be manually changed to **Active** when operation begins.

stack

A stack is an operational refinement of an application, component, or standard platform used in strategic planning. A stack represents the physically installed infrastructure that is defined in a deployment. Stacks may be hierarchically organized and contain multiple sub-stacks, thus allowing for various levels of operationalization.

At every level of refinement, the variability of the stack is further restricted. For example, where a first level stack distinguishes between operating system compatibility between UNIX and Windows in general, the next level of stacks may differentiate between various UNIX dialects or operating system versions.

For components that are created on the basis of vendor products, the stacks typically define the patch level release versions given by the vendor. For applications, the stacks are defined as a refinement of the application's platform. The stacks subordinated to the platform may exclude elements that had been used in the platform, thus representing complex group dependencies between the elements positioned in the platform tier - platform layer matrix. Additionally, stacks may introduce stack configuration items as new elements that have no relevance on the strategic level of the plan, but become important on the various levels of operationalization.

stack configuration item

A stack configuration item is a technical aspect of an application, component, or standard platform that is deemed unimportant for strategic planning purposes but relevant for operational and release planning as well as physical installation.

Stack configuration items are part of the technical description of a stack assigned to either an application, component, or standard platform. Stack configuration items themselves may reference a component. Typical examples for stack configuration items are install shields, installation scripts, job queues, etc.

stack element

A stack element is a constituent of a stack. Every stack element either references an element of the associated platform or is a stack configuration item. A stack element inherits the authorized user and/or authorized user groups from the stack that it is derived from.

stack upgrade

A stack upgrade describes the upgrade process to be defined from one stack to another stack. The stack upgrade is the result of the upgrade process. In other words, the stack upgrade is an object associated with the target stack. A stack upgrade has the same capabilities of a regular stack.

staff

A staff member is a person in an organization who is specified as a staff member to provide a specified skill.

standard component

See component

standard platform 

A standard platform is a platform that is defined outside of the scope of a concrete application or component. A standard platform aims to establish standards for the technical architecture. A common use case is to compare the concrete platform of an application to standard platforms in order to assess the overlap and hence the degree of standard compliance.

Standard platforms can be used to represent technical platforms (for example, "Standard Windows Desktop") as well as business platforms (for example, "Standard SAP BW Platform"). Much like any platform, the standard platform is based upon a platform layer and platform tier structure defined in a platform template.

standard platform category 

A standard platform category bundles and classifies content-specific standards. Standard platform categories allow you to hierarchically structure the standard platforms in your enterprise. Each standard platform may be associated with only one standard platform category.

standard platform element 

A standard platform element is the constituent of a standard platform. A standard platform element is associated with a standard component.

state

See object state

statutory language

A statutory language is a language that allows the enterprise to address regulatory requirements that mandate that objects are captured in a specific language. The statutory language may be the enterprise's primary language or a secondary language configured for the Alfabet user interface.

strategic business support

A strategic business support is a targeted business support defined for an unspecified timeframe in the context of the long-term target architecture. A strategic business support is typically provided by an existing or planned ICT object. However, strategic business support may also be configured to be provided by a virtual ICT object to define an ad-hoc ICT object that does not yet exist in the inventory, a solution building block to describe functional planning as endorsed by TOGAF, or an organization to describe non-IT support such as internal or external services.

swim lane

A swim lane is a boundary element in a service diagram that serves to separate different agents that perform activities. An agent can be a role played by a person or an application providing a business service that is requested by the business process visualized in the service diagram. Activities, events of the type Intermediate, and sequence flows can be placed within the boundaries of swim lanes. A message flow can connect activities from different swim lanes.

T

tactical business support

A tactical business support is a prescribed business support defined for a specified timeframe in the context of the medium-term to-be architecture. A tactical business support can be provided by an existing or planned application, solution building block, ICT object, or virtual ICT object, or in the case of support via internal or external services, an organization. The tactical business support provides support to either a business process (and business process version) or domain and can be provided to either an organization or market product.

target architecture

The target architecture represents the desired long-term status of the IT landscape without any specific timeframe or concrete references to applications.

A target architecture is described by an IT strategy. As such, several target architectures may be defined that differ for varying long-term business scenarios for the enterprise. The plan of action to achieve the currently applicable target architecture is represented in the master plan.

The target architecture usually resembles a strategic guideline rather than a detailed architectural plan. Therefore, the target architecture is described in less detail than the as-is architecture or to-be architecture. The description of the target architecture is usually restricted to the strategic business support and is formulated as a steady state without a period of validity or lifecycle description for its constituents.

technical architecture

See enterprise technical architecture

technical environment 

A technical environment describes a specific area of the technology required to support the development, maintenance, operation, or testing of an object. Technical environments are typically documented for testing and operations management purposes and may include, for example, Development Tools or Testing Tools. Technical environments could also be implemented to document the applications or components that are developed in-house. In this case, a number of design, development or testing tools (components) may be required that must be evaluated in the context of the application/component they are used for.

A technical environment definition can be configured for any of the following object classes: Application, Component, Deployment, Device, Standard Platform, and Vendor Product. If the object class supports object class stereotypes, any configured stereotypes may be configured to support the definition of technical environments. The technical environment definition also determines the user profiles that have Read/Write permission to specify the technical architecture elements relevant to the selected object.

Each technical environment definition can have multiple technical environment definition items and an unlimited number of levels in the technical environment definition item hierarchy. An Alfabet query is typically defined for a technical environment item in order to find the set of architecture elements that could be specified for an aspect of the technical environment.

For example, the technical environment Testing Tools could include the subordinate technical environment items Functional Testing, Performance & Load Test, etc. Each technical environment item would include an Alfabet query that would find the relevant set of testing tools that a user can choose from in order to define the object's testing environment. For example, an Alfabet query associated with the technical environment definition item Functional Testing might be configured to find the applications relevant to the functional testing of devices.

technical environment item 

A technical environment item describes an aspect of the technical environment that should be documented. The technical environment item typically includes an alfabet query that finds the set of architecture elements that could be specified for that aspect of the technical environment. Users with permissible user profiles can choose from the query's result set in order to define the architecture element that supports an object in the technical environment.

technical service

A technical service is a service provided by a component in order to fulfill technical needs that are necessary to support business service requests. A technical service may define technical service operations that support the technical realization of business functions.

Technical services can be defined for components and local components for which the **Component Type** attribute is set to `Service`. When a local component of the type `Service` is embedded into the

architecture of an application or another component (of the type `Infrastructure` or `Business`, for example), the technical service operations of the embedded component can be mapped to the business services provided by the application or the local component.

Object class stereotypes may be configured for the class **Technical Service**.

technical service operation

A technical service operation is a detailed description about how a technical service is to be provided by a component in order to support a component in providing a business service. An operation is defined for the specific technical service that it helps to realize. The operation defined for a technical service can be assigned to the business services that are provided by the component that the component supports. A technical service can have multiple technical service operations defined for it.

Object class stereotypes may be configured for the class **Technical Service Operation**.

technology 

A technology is anything using technical processes, methods, or knowledge in the IT landscape. Technologies can be structured in technology groups. Furthermore, technologies can be associated with other technologies. For example, the technologies ADO.NET and C# could be associated with the selected technology.NET since they are both.NET technologies.

A technology can be associated with multiple components, vendor products, and standard platforms thus indicating that these objects are instantiations of a certain technology. A component, vendor product, or standard platform may be associated with multiple technologies. Lifecycle analyses provide insight to the technology's lifecycle as well as the lifecycle of the components, vendor products, and standard platforms that the technology is used in.

Technologies may also be assigned to an IT capability thus indicating that the technology may potentially support the IT capability via one or more components.

technology group 

A technology group is a container to logically structure technologies. There may be various ways to logically structure technologies. Therefore, any technology may be associated with multiple technology groups. Object class stereotypes may be configured for the object class **Technology Group**.

technology support map 

A technology support map documents the technologies) and their associated components that realize an IT capability at one or more locations (typically data centers).

text template

- assignments
- workflows
- discussion groups
- consistency monitors, notification monitors, activity monitors, inactivity monitors, and date monitors
- notepad use in business support maps
- structural changes to the organizational model
- structural changes to the business process model

threat 

A threat refers to the source of a particular type of risk to the enterprise's IT. A threat can be associated with a specific risk object in the enterprise's IT landscape, indicating that the risk to the object is derived from the threat. Furthermore, risk mitigation templates can be defined for the threat in order to articulate one or more risk mitigations targeted to avoid, reduce or contain the risk derived from the potential threat.

tier

See platform tier

time series evaluation type

A time series evaluation type is an evaluation type for which target values can be defined for active time series periods for objects in the object class that the evaluation type is assigned to. Users can specify target values for the indicator types associated with the evaluation type. Multiple time series evaluation types can be assigned to an object class.

time series period

A time series period is a period of time (for example, 2010-01) for which a configured time series evaluation type can be evaluated. A time period can be activated and deactivated. Any time period that is activated will be visible in Alfabet views and users can thus evaluate the indicator type associated with the time series evaluation type for the objects in the relevant object class. Once a time period is deactivated it can no longer be evaluated and will no longer be visible in views.

to-be architecture

The to-be architecture is a medium-term plan for the enterprise's architecture and defines a clearly approved strategic guideline. The planned consolidated to-be architecture is captured in one or more master plans, which define the planned tactical business support. The realization of the to-be architecture is addressed in the context of projects.

A project's to-be architecture typically includes inventory objects from the as-is architecture that may be affected by proposed architectural changes as well as new solutions objects such as solution applications, solution components, solution devices, solution information flows, solution local components, solution peripherals, solution standard platforms, solution technical services, and solution business supports created in the context of the to-be architecture. The solution objects are considered to be "shadow" objects — copies of the base applications, information flows, etc. that they were derived from. The solution objects exist only in the context of the to-be architecture definition and may be modified within the context of the to-be architecture without affecting the inventory objects they are based on. The architectural changes defined in the context of the proposed to-be architecture can be checked in to the inventory, whereby the solution objects that are based on inventory objects overwrite their corresponding inventory object. All architecture elements added to the inventory via the proposed to-be architecture will have an object state indicating that the object is planned. The object state must be manually changed to an active object state when operational use of the to-be architecture element begins, which is typically upon completion of the project's implementation.

U

user

A user is a person that is known to the Alfabet system. Users have access permissions and are associated with one or more user profiles, which control accessibility to the functionalities. A user may or may not be an authorized user for an object. Authentication and authorization are also controlled by means of the user definition.

user group 

A user group bundles a set of users. Once a user group is assigned to an object, it is referred to as the object's authorized user group.

user profile

User profiles are the basis of user administration in Alfabet and serve as the entry point when accessing Alfabet. Every user must log in with a user profile that has been assigned to him/her by a user administrator. Therefore, all users accessing Alfabet must be assigned at least one user profile. However, users may possess multiple user profiles in accordance with their responsibilities in the user community and in the enterprise as a whole. A user can switch to another permissible user profile at any point during a user session.

A user profile specifies the Alfabet functionalities available to a user, the visibility and editability of object classes and object class attributes, as well as the availability of associated capabilities including, for example, wizards and workflows.

V

value node

A value node represents a strategic intention at a specific level of abstraction. The level of abstraction is determined by the value node stereotype that the value node belongs to. Value nodes assigned to a top-level value node stereotype represent a highly abstract strategic intention (for example, Become an Integrated Financial Service Provider) whereas the leaf-level value nodes typically describe the action needed to realize the ascendant strategic intention. Thus the value node stereotype at the lowest level usually represents strategic initiatives (for example, Establish New Credit & Loans Management Solution).

Value nodes assigned to adjacent value node stereotypes may be linked via a parent/child relationship. A value node may have multiple parent value nodes as well as multiple child value nodes. The relationship between two value nodes is represented by value node arcs, which allows for the ranking and relative importance of the strategic intentions to be computed. Value nodes can also be linked to architecture elements, thus describing which architecture elements may be impacted by a particular value node.

A perspective can be assigned to a value node in order to express a specific view that the enterprise has for the value node. Furthermore, multiple objectives can be defined for each value node in order to express a refinement of the strategic intention that can be acted upon. The assessment of measure types can be rated for the enterprise and targets can be defined for the objectives for an active time series evaluation type.

An organization may be specified as the owner of the value node, thus ensuring responsible for the realization of the strategic intention. A balanced scorecard analysis provides insight to the realization of the objectives associated with the value nodes according to the perspectives critical to the enterprise.

value node arc

A value node arc connects two value nodes of adjacent value node stereotypes and indicates that a subordinate value node is expected to contribute to achieving the strategic intention of its parent value node. The value nodes subordinate to a given parent value node are ranked to determine their relative contribution to realizing the parent value node.

A value node may be associated with multiple parent value nodes and thus ranked differently in each parent-child context. Each value node arc weight reflects the value node's contribution to the respective parent value node. Thus, a value node may have many different value node arc weights.

value node stereotype

A strategy network is made up of an arbitrary number of levels called value node stereotypes. The top-level value node stereotype represents the most abstract level of strategic intention and the lowest leaf-level

value node stereotype represents those strategic initiatives that must be acted upon in order to realize the abstract strategic intentions. For example, a typical strategy network might be made up of the following value node stereotypes: Level 1: Vision, Level 2; External Trends, Level 3: Business Drivers, Level 4: Business Requirements, Level 5: Architectural Requirements, Level 6: Initiatives. The name, number, and hierarchical ordering of value node stereotypes is configurable.

A value node is thus defined for a specific value node stereotype. Any pair of value nodes that are related to one another in the strategy network must belong to different value node stereotypes that are adjacent to each other in the hierarchy. In other words, value nodes are related via a parent-child relationship. The relationship between two value nodes is represented by value node arcs, which allow the relative ranking of the strategic intentions to be computed.

value stream 

A value steam describes the series of value stream steps that are relevant to implement solutions that provide a continued flow of value to the enterprise. Value streams allow "value" in terms of the usefulness, advantage, or benefit of IT solutions to be articulated and measured in the business architecture. Non-monetary examples of value are, for example, the successful delivery of a product or service or access to up-to-date information to make better business decisions. Value stream target values can be defined to articulate the strategic intentions (value nodes) in the enterprise's strategy network to be specified as the target that the value stream shall deliver value.

value stream step 

A value stream step describes each stage required for thevalue stream in order to deliver value to relevant stakeholders. The value stream step may be derived from a business process. The definition of the value stream step includes the stakeholder organizations and thebusiness capabilities or domains that are leveraged by the value stream step. For each value stream step,value stream step conditions can define the conditions required to enter and exit the value stream step. Indicators can be specified to capture the values that are delivered by the value stream step.

value stream step condition

A value stream step condition allows the conditions relevant to enter or exit a value stream step to be specified. The conditions are a semantic description of what must occur to enter or exit the value stream step. Evaluations can be defined for each condition in order to define indicators that measure the fulfillment of the condition.

vendor 

A vendor is a supplier of vendor products. A vendor product is marketed and sold by one single vendor. Object class stereotypes may be configured for the object class **Vendor**.

vendor group

A vendor group is a container to logically structure vendors. There may be various ways to logically structure vendors. Therefore, any vendor may be associated with multiple vendor groups.

vendor product 

A vendor product is a good or service that is provided by a vendor to the company. A vendor product may be associated with one or more technical components. For example, the vendor product Oracle RDBMS may be split into the components Oracle 11i Server, Oracle 11i Client, and Oracle 11i OCL. Further, a component may reference a vendor product, thus indicating that it has been derived from the vendor product.

Vendor products may be created directly in Alfabet or, depending on the configuration of your Alfabet solution, imported from the Technopedia® repository. Vendor products may only be imported from Technopedia® product categories that are on the leaf level of the product category hierarchy.

A vendor product is marketed and sold by one single vendor. A vendor product may be assigned to one vendor product category.

Object class stereotypes may be configured for the object class **Vendor Product**.

vendor product category 

A vendor product category bundles and classifies content-specific vendor products. Vendor product categories allow you to hierarchically structure the vendor products in your enterprise. Each vendor product may be associated with only one vendor product category.

vendor product group

A vendor product group is a container to logically structure vendor products. There may be various ways to logically structure vendor products. Therefore, any vendor product may be associated with multiple vendor product groups.

virtual ICT object 

A virtual ICT object represents a visionary ICT object that may not yet be defined in the inventory but is needed to represent a strategic business support or tactical business support defined in an IT strategy map or a master plan map. A virtual ICT object is for planning purposes only and is not visible anywhere outside of the areas of strategy planning or master planning.

After a virtual ICT object is defined in a business support map, it can later be replaced with a real ICT object or application. In this case, the real ICT object replaces all business supports associated with that virtual ICT object.

virtual organization

A virtual organization is either a temporary or permanent organization such as a decision board, steering committee, or other bodies that exists outside the formal organizational structure of an enterprise. For example, a virtual organization could be an IT committee that provides strategic direction to the enterprise in terms of the alignment of business and IT.

W

wizard

A wizard is an assistant that consists of a configured set of wizard stepsthat may include standard editors, custom editors, standard views, and configured reports. The wizard will typically guide a user through a linear multi-step process to capture data for an object that the user has access permissions to.

Standard data capture wizards are available for commodity classes in which a large number of objects are typically documented. An unlimited number of custom wizards may be configured for an object class in order to allow different users working in different contexts to capture the object data that they are responsible for. However, only one wizard may be available for an object class per user profile.

wizard step

- a standard editor
- a custom editor with one tab
- a custom editor with multiple tabs
- a custom editor with visibility rules that determine the visibility of custom editor pages based on the values defined for a specified custom object class property
- a standard page view
- a tabular query-based custom report
- a query-based custom report displaying graphics (such as a tree-map or layered diagram report)

workflow

A workflow is a collaborative process made up of workflow steps that are typically carried out by one or more users. A workflow is based on a configured workflow template that determines the sequence of workflow steps that are to be performed on a specific object and its references by specified user(s). Workflow steps may have specific pre- and post-conditions that determine different paths to take in the workflow depending on whether the conditions are or are not met.

Typically, the workflow owner is the user who initiates and is responsible for maintaining the workflow. When a workflow is initiated by the workflow owner and when a workflow advances to the next workflow step, relevant users may be informed via automatically-generated emails of their impending responsibility for the workflow step. This functionality ensures that all relevant users are informed and reminded of their responsibilities in the collaborative workflow. The option to refuse, delegate, and pause workflow steps, remind users of an impending target date for a workflow step as well as redirect a workflow that has encountered an error enables workflow owners and workflow administrators to keep track, coordinate, and manage the completion of each workflow step and the workflow as a whole.

workflow step

A workflow step is an activity or task in a workflow that must be performed. The workflow template may have as many workflow steps as needed to complete the relevant task. Standard and custom editors, wizards, page views, configured reports, and object profiles may be implemented in the workflow step so that users can complete the task. A workflow step may entail entering, modifying or reviewing data, or a workflow step may be configured to be automatically executed by the Alfabet system. If the workflow step is not automatically executed by the system, one or more users may be defined as responsible to perform the workflow step. A user may delegate a workflow step to another user or refuse a workflow step. Depending on the configuration of the workflow step, a user may be required to confirm a workflow step before the workflow can advance to the next workflow steps, or the confirmation of the workflow step may occur automatically. A workflow step may also have one or more associated workflow step actions which allow various operations to be performed when the workflow step is entered, exited, refused, or expired. Additionally, each workflow step may have one or more pre-conditions or post-conditions that must be fulfilled in order to enter or exit the workflow step.

workflow template

A workflow template is a customer-defined blueprint for one or more workflows. The template specifies what object class is the point-of-departure in the workflow, which user groups and/or user profiles may initiate and administrate the workflow, which workflow steps comprise the workflow as well as their sequence, any possible per- and post-conditions or update actions associated with a workflow step, and what kinds of workflow notifications should be sent to collaborating users in which contexts. The user who creates the workflow template is the workflow template owner.

A workflow template must have the attribute **Workflow State** attribute set to `Plan` to be configured and validated. Once the workflow template is completed and approved, the attribute **Workflow State** must be switched to `Active` in order to make it available to the user community. Once the workflow template is available in Alfabet, a permitted user may initiate a workflow based on the workflow template. Multiple workflows may be simultaneously initiated and running for a workflow template.

X

Y

Z