1.1 Requirements Model Engineering

1.1.1 EAMF Framework and Methodology Aspects

The benefit of using the EAMF methodology is to help Business Architects work with Business Analysts to create a structured and cohesive requirements model that can serve as a foundation for the various architectural perspectives set forth in the EAMF framework. This foundation will guarantee that all architects are on the same page hereby eliminating the classical disconnect between business and technical project staff. The corresponding requirements model structure is described in Section 1.1.1.1 and the relationships between requirements and definitions, including traceability, are discussed in Section 1.1.1.2.

To establish a cohesive foundation, Business Architects follow a two-step iterative approach. First, Business Architects need to map existing requirements into corresponding requirements and/or definitions within the EAMF requirements model illustrated on the right side of Figure 1. The actual sub-steps followed by Business Architects are detailed in Section 1.1.1.3. The second step consists in analyzing requirements using a reasoning approach about business entities and their dependencies/goals that leads to more precise requirements and/or definitions. Corresponding sub-steps are documented in Section 1.1.1.4. Business Architects may need to iterate on these two steps to create a cohesive requirements model that leverages the initial set of requirements obtained from Business Analysts. Throughout these steps, and as they analyze the requirements, Business Architects strive to explain and document why they are structuring the requirements model in a specific way.

1.1.1.1 Project Requirements Model Categories

The EAMF requirements model is compiled by Business Architects according to the template shown in Figure 1 below. The requirements model is partially based on requirements captured by Business Analysts during the requirements engineering phase. Figure 1 details requirements types and definitions used in the Project Requirements Model Categories. In addition, there are several EAMF Catalogs and Enterprise Requirements Model Categories that are mentioned at the bottom of Figure 1. The Enterprise Projects category uses the same template as the Project Requirements Model category. The other categories use additional and/or different requirements and definitions types and have their own separate templates. It is important to note that Business Architects may decide to introduce new requirements and definitions categories and/or requirements and definitions types on a project per project basis and/or to satisfy a specific need identified at the Enterprise level. In general, the terminology used in this document conforms to the definitions set forth in the Business Motivation Model paper [BMM] published by the Business Rules Group (BRG).
The Requirements Model categories are split into two parts. The first part includes definitions that Business Architects decide to adopt based on their analysis of existing requirements and or previous business architecture knowledge obtained from the Enterprise Requirements Model categories. The second part includes all additional requirements that are formulated by Business Architects as a result of reasoning about business entities and their dependencies/goals based on adopted definitions. These additional requirements result from analysis input motivated via reasoning. To come up with these additional requirements, Business Architects analyze existing definitions as explained in Section 1.1.1.4 and come up with new ones as needed to help elicit
requirements. These new definitions are added to the categories in the first part of the Requirements Model. For example, as new business participants are created, they are modeled as new actors and corresponding definitions are added to the Project Business Vocabulary Definition category. These actors are then considered as new requirements and placed in the Business Entity Requirements category. At that point they become candidate requirements for the Organizational Business Model Requirements category. The input to the reasoning process that creates requirements for the second part of the Requirements Model is, in fact, the set of definitions that are in the first part. Project Requirements Model Categories and EAMF Catalogs/Enterprise Requirements Model Categories are further defined as follows:

**Project Requirements Model Categories – Part I: Definitions:**

The *Project Business Vocabulary Definition* category is used to hold definitions of terms that must be used in a project. These terms may reference Glossary and Enterprise Business Vocabulary requirements created by Business Analysts. Additional terms may be references to vocabulary definitions contained in the Enterprise Glossary that is part of the Enterprise Requirements Model Category. New definitions will also be added as Business Architects reason about business entities and their dependencies/goals. An example of a Project Business Vocabulary Definition may be a “Service Request Processor” defined as “A business entity that stores and retrieves service requests”.

The *Strategy Definition* category is used to hold definitions of high-level business needs that must be addressed by a project. Strategy Definitions may reference Stakeholder Requirements and Business Strategy and Innovation requirements created by Business Analysts. Additional definitions may be references to strategies definitions contained in the Enterprise Strategies Category that is part of the Enterprise Requirements Model Category. Strategies usually channel efforts towards high-level business goals (i.e., objectives). An example of a strategy definition may be: “Notification letters must be sent to the location of the customer’s choice”.

The *Concept Definition* category is used to hold definitions of Project Business Directives, Business Objectives, and Features and Events that must be addressed by a project. Project Business Directives definitions state Business Policies. These definitions may be references to Business Policies Requirements created by Business Analysts in the Business Rules Requirements Repository. Additional definitions may be reference to business policy definitions contained in the Enterprise Business Rules Category that is part of the Enterprise Requirements Model Category. An example of a Project Business Directive definition may be: “Policy renewals must be allowed via the Internet”. Business Objectives definitions typically reference Business Objectives requirements created by Business Analysts. An example of a Business Objective definition may be: “Automated channels for customer service requests processing must be provided”. Features and Events definitions typically reference Features and Events requirements created by Business Analysts. An example of a Feature and Events definition may be: “Change request submitted by the employee/member”.

**Project Requirements Model Categories – Part II: Requirements:**

The *Business Entity Requirements* category is a placeholder for a set of references to participants and tasks that are initially identified by Business Architects while reasoning about business entities and their dependencies/goals based on the definitions they adopted in the first part of the Requirements Model. As will be explained in Section 1.1.1.4, that reasoning process leads to a
representation of Business Model\textsuperscript{1} Requirements in terms of actors, goals of actors, and dependencies between actors in achieving these goals. As goals are refined into high-level tasks, the reasoning process leads to a representation of Business Use Case Requirements in terms of high-level tasks that become the responsibility of actors and allow them to achieve their goals. While the Business Entity Requirements type may seem to overlap with the Organizational, and Location types (included in the Business Model Requirements category) and the Business Use Case Requirements type, there are two reasons that justify its presence in the Requirements Model Categories. The first reason is linked to the reasoning process that will be explained further in Section 1.1.1.4. In a nutshell, as part of that reasoning process, Business Architects may come up with a few analysis alternative design choices supported by different sets of Business Entity Requirements. Procedurally, it makes sense at first to keep track of the overall set of Business Entity Requirements in a separate category until further refinements are conducted as part of the reasoning process. The second reason is that having all Business Entities Requirements in a separate category reduces duplication and facilitates their referencing from Business Use Cases, and Organizational/Location Requirements, pertaining to multiple sets of Business Model Requirements. Each set of Business Model Requirements corresponds to an alternative representation that results from the aforementioned reasoning process. As explained earlier, when Business Architects come up with new actor definitions as part of the reasoning process, they are also added to the set of Project Business Vocabulary Definitions. However tasks are not documented in the Project Business Vocabulary Definitions category. An example of a Business Entity Requirement may be a reference to “Service Requestor” that would be defined in the Project Business Vocabulary Definition category as “A party who places a service request and consumes the results of the request”.

The Business Use Case Requirements\textsuperscript{2} category hosts special kinds of high-level functional requirements. These functional requirements are supported by one or more high-level tasks that convey how a business solution should interact with users (called actors) to achieve a specific business goal or function. These scenarios are initiated by external events or may encapsulate processes that must be followed to take advantage of a feature. In short, a Business Use Case captures who (actor) does what (interaction) with the business solution, for what purpose (goal), without dealing with business solution internals. Business Use Case Requirements typically reference Use Cases Requirements created by Business Analysts. Business Architects analyze Use Cases Requirements further based on tasks identified via the reasoning process described in Section 1.1.1.4. Their goal is to create very essential (i.e., high-level) use cases, add new underlying tasks, and improve the grouping of underlying tasks. An example of a Business Use Case Requirement derived from the event “Change request submitted by the employee/member” may be: “Handle change request received from employee/member”.

\textsuperscript{1} The EAMF methodology adopts the widely accepted definition of a business model set forth in [PLACE] as “a set of participants and the flows between them”. The participants include the company behind the business model of interest and all the actors involved. Value is created in the form of information, product, and money flows between the participants. These various types of flows are generally expressed in abstract terms as dependencies between actors.

\textsuperscript{2} Note that upfront definitions of Business Use Case Requirements are not required to apply the EAMF methodology. The EAMF Business Architecture Analysis and Design steps are generally applied to the set of requirements definitions categories that are in the first part of the Project Requirements Model. While the EAMF methodology may leverage existing Business Use Case Requirements, following the steps in the EAMF Requirements Engineering and Business Architecture Analysis and Design disciplines typically results in a new or refined set of Business Use Case Requirements that better describe how a business solution should interact with users to achieve stated business goals or functions.
The Business Model Requirements category includes the Location, Organizational, and Process Model Requirements that must be addressed by a project. Process Model Requirements specify the Business Process Requirements, Business Rules Requirements, and Business Workflow Rules Requirements that must be addressed by a project. The definitions of these requirements types are the same as the definitions given for Business Model, Location, Organization, Process, Business Rules, and Business Workflow Rules requirements respectively. These requirements definitions may reference corresponding requirements created by Business Analysts. However, additional requirements are always formulated by Business Architects for the Organizational and Location sub-categories as they reason about business entities and their dependencies/goals based on the definitions they adopted in the first part of the Requirements Model. The reasoning process and its expected outcome are described in detail in Section 1.1.1.4.

The Requirements Definition category is used to hold Functional and Non-Functional Requirements for a project. Functional Requirements constrain the behavior of a business solution and may be expressed in terms of services, tasks or functions that the business solution must perform. Functional Requirements stored in the Requirements Model typically reference Functional Requirements created by Business Analysts. An example of a Functional Requirement may be: “An employee shall be able to submit a name change request”. Non-Functional Requirements specify qualitative constraints that should be addressed by a project. Non-Functional Requirements cover both interface and operational constraints placed on a business solution and are categorized in the same way as the corresponding requirements shown in Error! Reference source not found.. Non-Functional Requirements typically reference Non-Functional Requirements created by Business Analysts. Non-Functional Requirements include non-functional project, process, and external requirements. Non-functional project requirements correspond to quality metrics that should be met by the business solution being developed as part of a project. Examples of such metrics include reliability and efficiency. Non-functional process requirements pertain to the guidelines that should be followed while working on a project within the Enterprise. An example of a non-functional process standard requirement may be: “Standards for business rules and business policy shall abide to the specification of directives specified in Revision 1.2 of the OMG’s Business Motivation Model specification”. Non-functional external requirements capture external constraints that are imposed on a project such as industry regulations. An example of a non-functional external HIPAA regulatory requirement may be: “A service requestor shall always be identified”.

EAMF Catalogs:

The Pattern Catalog category, also referred to as the EAMF standard pattern catalog, contains definitions of pattern cluster networks and associated pattern languages. Examples of such definitions are the descriptions of various types of business service patterns or agent patterns. Business service patterns include Enterprise Reference Elements such as Component Services, Composite (Business) Services, and Conversational (Workflow) Services\(^3\). Agent patterns are further categorized as architecture, communication, travel, or coordination agent patterns. Placing

\(^3\) Component services (also called technical or application services) support simple atomic actions on a subset of business related entities that do not depend on another service to function. Composite (Business) Services, also called business services, are also atomic in nature, but orchestrate the invocation of component services into a business level process. A composite service is stateless (unlike the workflow service), does not manage a long lived transaction (again, unlike the workflow service), and may be invoked synchronously or asynchronously. A conversational service (also called workflow service) typically has state attached to it and looks like a classical finite state machine. BPEL4WS type Web service composition would be an example of a conversational Web service.
EAMF pattern definitions in the Pattern Catalog category allows individual projects to relate to them in this common category which facilitates the reusability of pattern definitions and avoids their duplication.

The Product Catalog category references definitions of products contained in the EAMF standard and domain-specific product catalogs. Examples of such definitions are the descriptions of various Enterprise Services Buses (ESBs) products. Placing EAMF product definitions in the Product Catalog category allows individual projects to relate to them in this common category which facilitates the reusability of product definitions and avoids their duplication.

The Enterprise Solution Catalog category, also referred to as the EAMF domain-specific pattern catalog, contains definitions of pattern cluster networks and associated pattern languages. Examples of such definitions are the descriptions of various types of agent as delegates patterns used in eCommerce architectures. Placing EAMF Enterprise solution definitions in the Enterprise Solution Catalog category allows individual projects to relate to them in this common category which facilitates the reusability of Enterprise solution definitions and avoids their duplication.

**Enterprise Requirements Model Categories:**

The Enterprise Glossary category contains glossary definitions used to specify terms that are commonly used in the Enterprise and must be used across all projects. Note that there may be project specific Business Vocabulary Definitions that are not part of the Enterprise Glossary. These project specific glossary definitions are typically maintained within individual Project Requirements Models. An example of a glossary definition that falls into the Enterprise Glossary category is an “Account” defined as a “Group of employees each of which has at least one policy”. Placing glossary definitions in the Enterprise Glossary category allows individual projects to relate to definitions in this common category for reusability purpose rather than duplicate glossary related definitions.

The Enterprise Business Rules category contains business rules, business workflow rules, and business policies definitions. These definitions may be used to govern the behavior of business processes that pertain to specific projects. Higher level Enterprise tactical requirements that implement strategic requirements may regulate the enforcement level of individual business rules. Enterprise business rules (and associated requirements) may also be the basis for a hierarchy of business policies and associated requirements that govern Enterprise business processes. Enterprise workflow rules definitions state the process by which business is done and are bound by the Enterprise business rules and/or policies definitions. Enterprise workflow rules abide to the business workflow rules definitions contained in the Enterprise Business Rules category. Note that there may be project specific business and workflow rules definitions that are not part of the Enterprise Business Rules category. These project specific definitions are typically maintained within individual Project Requirements Models. An example of a policy definition for an insurance company may be: “The company must contact a policy holder to verify the validity of a service request (e.g., add person to a policy) when a non-policy holder places a service request on his or her policy”. The associated business rule definition may be: “A policy holder must approve any service request issued on his or her policy”. The corresponding workflow rule definition may be: “When a service request is placed on a policy by a non-policy holder, the company must send a letter to the policy holder to verify the validity of a request and wait for a signed letter to come back from the policy holder before processing the service request”. Placing business rules and business workflow rules in the Enterprise Business Rules category allows individual projects to relate to definitions in this common category for reusability purpose rather than duplicate business rules and business workflow rules related definitions.
The *Enterprise Solution Patterns* category classifies project governance (i.e., process pattern) definitions sub-categories that are maintained by the Project Management Office (PMO). An example of an Enterprise Solution Pattern sub-category is the SDLC definition sub-category used to classify functional and non-functional process definitions that support iterative and non-iterative SDLCs. An example of a functional process definition that falls in this category may be: “Consider all agent patterns individually when reasoning about business entities’ dependencies and goals”. Placing governance requirements in the Enterprise Solution Patterns category allows individual projects to relate to definitions in this common category for reusability purpose rather than duplicate project governance related definitions.

The *Enterprise Strategy* category contains various categories of definitions of high-level strategic and tactical stakeholder requirements that are driven by the company’s Strategic Planning Office. Business Strategy and Innovation requirements definitions are clearly above and beyond specific project requirements definitions. An example of a Business Strategy and Innovation Requirements definition sub-category is the project roadmap sub-category that classifies requirements definitions that apply to programs and projects and includes traceability all the way back to the company’s leadership strategic requirements. Placing high-level stakeholder requirements definitions in the Enterprise Strategy category allows individual projects to relate to definitions in this common category for reusability purpose rather than duplicate high-level stakeholder related definitions.

The *Enterprise Projects* category classifies requirements definitions sub-categories that pertain to Enterprise-wide solutions. An example of a requirements definitions sub-category that falls in that category is the Enterprise Worker Services (EWSs) requirements definitions sub-category that classifies requirements definition sets that apply to an extensible set of worker services (e.g., service lifecycle management) for Enterprise SOA. Placing a requirements definition set in the Enterprise Projects category allows individual projects to relate to definitions in this common category for reusability purpose rather than duplicate high-level Enterprise project requirements definitions.

In addition to the categories listed in the bottom right corner of Figure 1, an example of a custom Enterprise Requirements Model category that may be useful to Business Architects would be an *Enterprise Project Assessment Definition* category. This custom category would be used to hold projects-to-be requirements definitions as part of a feasibility study.

### 1.1.1.2 EAMF Requirements Model Relationships

The various types of relationships that may exist between project requirements and Requirements Model definitions/requirements are defined in this section. The detailed steps followed by Business Architects during requirements model engineering, including the documentation of relationships, are explained further in the following two sub-sections.
Engineering activity. These suggested traceability relationships should be used and captured by Business Architects when mapping existing requirements created by Business Analysts to requirements and definitions in the Requirements Model. From a requirements change management standpoint, it is the joint responsibility of Business Analysts and Business Architects to maintain the accuracy of traceability relationships between Project Requirements and Requirements Model Definitions and Requirements.

**Relationship Between Requirements Model Definitions and Requirements:**

The requirements and definitions traceability graph illustrated in Figure 2 documents traceability relationships that may exist between the different types of requirements and definitions listed in the Project Requirements Model categories shown in Figure 1. Bidirectional arrows indicate that trace to or trace from (but not both) traceability relationships may be created between requirements and/or definitions types being pointed to. Business Architects should only document direct traceability relationships. Project Business Vocabulary Definitions are not used in the traceability graph but they are implicitly attached to the requirements and definitions set. It is the implied responsibility of Business Architects to apply glossary definitions throughout a requirements and definitions set.

Business Architects typically capture traceability relationships once they complete the mapping of existing requirements created by Business Analysts to requirements and definitions in the Requirements Model. These traceability relationships conform to those defined by Business Analysts between corresponding project requirements. From a requirements change management standpoint, it is the responsibility of Business Architects to maintain the accuracy of traceability relationships between Requirements Model Definitions and Requirements based on existing traceability relationships between project requirements.

As Business Architects reason about business entities and their dependencies/goals, more precise requirements and/or definitions will emerge and additional traceability relationships will be created as needed. It is the responsibility of Business Architects to maintain the accuracy of traceability relationships established as a result of the elicitation of new requirements and/or definitions based on the aforementioned reasoning process.
1.1.1.3 Mapping Requirements into the EAMF Requirements Model

The main goal of Business Architects involved in using the requirements model engineering discipline is to work with Business Analysts to collect and analyze existing requirements. They must then map and crystallize requirements into definitions, and reason about these definitions and/or requirements to assemble a structured cohesive requirements model that may serve as a foundation for the various architectural perspectives set forth in the EAMF framework. Business Architects must also keep track of traceability and/or hierarchical relationships between definitions and/or requirements and ensure that the set of definitions and requirements is complete. A set of definitions and requirements is complete when all the definitions and requirements below the Concept Definitions can be traced back indirectly to Strategy Definitions. Finally, Business Architects must collaborate with Business Analysts to keep the Requirements Model in sync with evolving Project Requirements as needed.
The first series of iterative steps that should be followed by Business Architects to meet the goal outlined above consist of mapping existing requirements into the EAMF Requirements Model. These steps are described below as steps 1-7. Note that steps 5-7 may be worked on in parallel. These steps assume a generic set of requirement/definition (and document) types that should be documented by Business Architects in a Requirements Model management plan. All requirements and definitions collected during a project increment should be reflected, either physically or via traceability, in the project Requirements Model Specification. The project Requirements Model management plan should be included with the Requirements Model Specification (either as a section or an addendum) and the Business / Information / Application / Technology / Enterprise Architecture Specifications to compile the report required to obtain sign-off from the Architecture Review board at the end of the project elaboration phase.

It is the responsibility of Business Architects to leverage requirements and definitions contained in the Enterprise Requirements Model categories. Early Requirements Model Engineering efforts should reference specific definitions from the EAMF Pattern and Product catalogs, and leverage Enterprise Solution Patterns defined in the corresponding Enterprise Requirements Model category. Business Architects may also leverage Strategy Definitions from the Enterprise Strategies category and policies/business rules/business workflow rules definitions from the Enterprise Business Rules category. Furthermore, all Enterprise Requirements Model categories (particularly the Enterprise Glossary, and Enterprise Projects categories) should be leveraged by Business Architects on an ongoing basis as part of the iterative Requirements Model elicitation process that is an inherent part of the overall Requirements Model engineering effort on a given project.

It is typically the case that projects are a continuation of work accomplished as part of past projects. In that case, Business Architects must verify that Requirements Model engineering work on the new project leverages requirements and definitions from previous related projects. It is the responsibility of Business Architects to (re)create requirements and definitions documentation for past projects as needed to facilitate Requirements Model engineering work on the new project as per the methodology described in this document.

**Step 1 - Gather Existing Requirements Documentation**

It is the responsibility of Business Architects to gain access to the latest project requirements set developed by Business Analysts. The requirement set must contain requirements traceability information. Joint efforts between Business Analysts and Business Architects should facilitate the mapping of the requirements set into a requirements model and should result in a finalized requirements specification. To facilitate the mapping steps, Business Architects should gain access to the electronic version of the requirements set. Business Architects must have access to

---

4 These combined set of specifications document the results of Architectural Analysis and Design steps in all architectural perspectives. The content of these specifications is discussed in later sections of this document.

5 Typically, Business Architects first obtain High-Level Requirements from Business Analysts, and put together a Requirements Model. Then they proceed with High-Level Analysis and pass their results back to Business Analysts to help them compile detailed level requirements that are included in a final requirements specification. This final requirements specification is used by Business Analysts to obtain stakeholder signoff. Business Architects update the Requirements Model based on the information contained in the final requirements specification and proceed with detailed level analysis and design. Business Architects document their detailed level design in a Requirements Model specification and obtain signoff from the Architecture board (which is part of the PMO).
all relevant requirements and/or new or updated requirements before proceeding with the remaining steps below.

**Step 2: Mapping Glossary Requirements into Project Business Vocabulary Definitions**

Existing Glossary requirements and references to Enterprise Business Vocabulary requirements created by Business Analysts are used as Project Business Vocabulary Definitions in the Requirements Model. Business Architects may decide as to their applicability. New definitions may be added by Business Architects upon review of the Enterprise Glossary definitions. Additional definitions are the result of the requirements elicitation process described in Section 1.1.1.4 as documented in Section 1.1.1.6. These definitions typically pertain to new business actors (and Resources) that are discovered while reasoning about business entities and their dependencies/goals.

**Step 3: Mapping Stakeholder Requirements into Strategy Definitions**

Stakeholder Requests requirements and references to Business Strategy and Innovation Requirements captured by Business Analysts are used as Strategy Definitions in the Requirements Model. Business Architects may decide as to their applicability. New definitions may be added by Business Architects upon review of the Enterprise Strategies definitions.

**Step 4: Mapping Existing Related Requirements into Concept Definitions**

Existing Business Objectives and Features and Events requirements captured by Business Analysts are used as Business Objective and Feature and Events definitions in the Requirements Model. Business Architects may decide as to their applicability. New definitions may be added by Business Architects upon review of the Enterprise Glossary definitions. Business Policies Requirements created by Business Analysts in the Business Rules Requirements Repository are used as Project Business Directives definitions in the Requirements Model. Additional definitions may be references to business policy definitions contained in the Enterprise Business Rules Category that is part of the Enterprise Requirements Model Category.

**Step 5: Mapping Existing Related Requirements into Functional and Non-Functional Requirements**

Existing Functional Requirements and Non-Functional Requirements captured by Business Analysts are used respectively as Functional Requirements and Non-Functional Requirements in the Requirements Model. As documented in Section 1.1.1.6, additional requirements may result from the requirements elicitation process described in Section 1.1.1.4. New requirements may also be incorporated in later iterations as Business Architects explore Business Model Requirements alternatives.

**Step 6: Mapping Existing Use Case Requirements into Business Use Case Requirements**

Existing Use Cases Requirements captured by Business Architects are the basis for Business Use Case Requirements in the Requirements Model. Business Architects typically analyze Use Cases Requirements further based on tasks identified via the reasoning process described in Section 1.1.1.4. This helps them create very essential (i.e., high-level) use cases referred to as Business Use Case Requirements. Business Use Case Requirements that result from this analysis may include new underlying tasks. Business Architects may also group the underlying tasks suggested
by Business Analysts in a different way as they define Business Use Case Requirements based on groups of tasks.

**Step 7: Mapping Existing Related Requirements into Business Model Requirements:**

Business Model, Location, Organizational, Process, Business Rules, and Business Workflow Rules requirements captured by Business Architects are used respectively as Business Model, Location, Organizational, Process Model, Business Rules Requirements, and Business Workflow Rules Requirements in the Requirements Model. Additional requirements placed in these sub-categories during Requirements Model Engineering are the result of the requirements elicitation process described in Section 1.1.1.4.

1.1.1.4 **Reasoning about Business Entities’ Dependencies and Goals**

As explained in Section 1.1.1.1, Requirements Model categories are split into two parts namely the definitions part and the requirements part. New requirements in the requirements part of the Requirements Model are derived from definitions created by Business Architects in the definition part of the Requirements Model as a result of mapping project requirements into the EAMF requirements model. The EAMF methodology provides a reasoning approach that helps elicit requirements in the requirements part of the Requirements Model. This reasoning approach is based on EAMF process patterns that originated from research applied to various contexts [BMDE][DEBMU][FRAD][SARMI*][SOARE][TTSDM]. As a result of applying this reasoning approach new definitions may come up and need to be added to the definition part of the Requirements Model. This section describes this reasoning approach and its outcome.

Figure 3 illustrates the details of the approach that consists in reasoning about business entities and their dependencies/goals. This approach consists of two steps as described below. As an alternative to the steps described in this section, a repeatable development process⁶ for conceptual models such as the ones presented here is documented in [TTSDM].

---

⁶ The suggested process is based on an iterative, actor and goal based, refinement algorithm. The algorithm assumes that a complete list of actors has been identified along with a list of associated root goals (and soft-goals). Each root goal is then analyzed from the perspective of its respective actor, and as sub-goals are generated, they are delegated to other actors, or the actor takes on the responsibility of dealing with them itself. The analysis is carried out concurrently with respect to each root goal. Sometimes the process requires the introduction of new actors and goals and/or tasks are delegated to them. The process is complete when all goals have been dealt with to the satisfaction of the actors who were initially liable for them (or the designers thereof).
Step1: Capturing Participants Requirements and Flow Requirements Between Them

As mentioned earlier, the EAMF methodology defines a business model as “a set of participants and the flows between them” as per [PLACE]. Based on this widely accepted definition and as illustrated in Figure 3, Business Model Requirements are typically motivated by value flows between participants where participants include the company behind the business model of interest and all other actors involved. Value may be created in the form of information, product, and money flows between participants. However, as Business Architects reach this step of the methodology no precise Business Model Requirements are available to them. All what is generally available is a set of high-level business requirements such as Business Objectives and other Informal Requirements definitions. These are in fact the definitions that were created by Business Architects in the definition part of the Requirements Model as a result of mapping project requirements into the EAMF requirements model. Therefore the conjecture of the reasoning approach is to provide a way to elicit Business Model Requirements (i.e., Organizational, Location, and Process Model requirements) given a set of High-Level Business Requirements.

While the above problem does not appear to be easily solvable directly, the EAMF reasoning approach relies on a simplified representation of Business Model Requirements that makes it easier to elicit, analyze, and evolve such requirements given a set of high-level business requirements. This simplified representation is the result of applying EAMF model-driven requirements elicitation process patterns7 to the aforementioned problem conjecture. Basically, to

7 EAMF promotes the use of model-driven techniques to drive the delivery of requirements by providing specific models of the type of information to be gathered. The model of choice depends on the nature of the
handle this problem, EAMF model-driven requirements elicitation\textsuperscript{8} process patterns suggest that business model requirements that are constrained by high-level business requirements may be represented in terms of: actors, actor goals, and dependencies between actors in achieving these goals. In this representation, actor goals are used to model high-level business requirements, business model requirements are represented in terms of actors and their dependencies, and dependencies indicate value flows between actors as illustrated in Figure 3. New actors are introduced whenever existing actors do not have the capability of satisfying their goals. In that case, dependencies on other actors must be created. Resources, which are not shown in Figure 3, may be introduced as needed to give actors the ability to satisfy their goals. This representation is simplified as it does not address the modeling of business processes but rather promotes their separation\textsuperscript{9} from the actors in the business model that conducts them. EAMF defers the elicitation of business process requirements to the Business Architecture Analysis discipline covered later. In summary, introducing goal, actor, and dependency modeling constructs allows Business Architects to capture participants requirements and flows requirements between them which is sufficient to identify business model requirements at a high-level (i.e., without delving into the details of associated business processes).

Note that the circled number “1” that appears twice on Figure 3 is there to indicate that the mapping of high-level requirements to goals and the modeling of high-level business requirements in terms of actors, actor goals, and dependencies must be performed at the same time so that goals are assigned to actors and the dependencies between actors become goal achievement dependencies.

To leverage the simplified representation of Business Model Requirements in terms of knowledge level concepts such as actor, goal, dependency, resource, and capability, EAMF suggests the use of conceptual models and representative diagrams developed as instances of these concepts. The type of conceptual model applicable to this first step is referred to as an \textit{Actor and Dependency Model}. This type of model results from the high-level analysis of actors, including their goals and goal achievements dependencies. Business Architects may document multiple Actor and Dependency Models. However, it is their responsibility and analysis choice to select one of these alternatives to best support the business architecture they are creating.

\begin{tabular}{l}
problem and the requirements under study. In general, requirements to be elicited include problem boundaries (i.e., what problem needs to be solved), stakeholders and user classes, goals and tasks, and scenarios and use cases. When high-level goals are available, EAMF process patterns promote the use of a goal-oriented model-driven technique that focuses on modeling the coordination of stakeholders to achieve these high-level goals. When it appears difficult to articulate requirements (including high-level goals) EAMF promotes the use of scenario-based model-driven techniques to elicit information about the tasks that users currently perform or might want to perform. \\
\end{tabular}

\begin{tabular}{l}
\textsuperscript{8} It is the authors’ experience that non model-driven requirements elicitation techniques are less formal, non-repeatable, and more demanding in terms of time and resources. These alternative techniques include traditional elicitation (i.e., via questionnaires, surveys, interviews, etc.), group elicitation (i.e., RAD/JAD workshops), prototyping (i.e., use of a prototype to motivate stakeholders’ feedback), cognitive elicitation (i.e., provide insights into the cognitive processes required to perform a task), contextual elicitation (i.e., a recent alternative to traditional and cognitive elicitation that emphasizes the identification of patterns in conversation and interaction). \\
\textsuperscript{9} This separation of concerns is very handy when designing business architectures as it allows the exploration of alternative business models while preserving investments in existing business processes. As documented in Section \textit{Error! Reference source not found.}, exploring alternative business models using Actors and Dependencies and Goal and Task Models can be done without changing underlying business processes.
\end{tabular}
EAMF process patterns do provide additional guidance to formalize the introduction of new actors in Actor and Dependency Models. This additional guidance originated from research focused on understanding and classifying agent\textsuperscript{10} patterns [PLMUA][PDDAS][RPAC] and business services \textbf{Error! Reference source not found.}. Note that Figure 3 does indicate that EAMF process patterns must be applied to the representation of Business Model Requirements in terms of actors, goals, and dependencies. In a nutshell, these process patterns suggest that there are several known agent and business services patterns, and related pattern language that Business Architects should try to apply as they reason about business entities dependencies and goals. These patterns typically apply to (groups of) business solution actors and govern the dependencies between business solution actors within a group. Business solution actors are considered as (generic) reference elements in the representation of business architectures in terms of actors, goals and dependencies.

As a starting point\textsuperscript{11}, EAMF leverages the pattern languages presented in [PLMUA] and \textbf{Error! Reference source not found.}. to model business solution actors and groups of actors (i.e., societies of actors) using agent or service patterns rather than plain actors. The structure of the suggested pattern language for agent patterns is illustrated in Figure 4 below. Model-driven agent-based goal-driven techniques have been used extensively for early requirements analysis [AOSD]. While agent concepts may be used to express conflicting human interests within a social system, they may also be used to model and reason about the distribution of capabilities and allocation of responsibilities within a physical business solution so that it becomes clear as to how computational elements are meant to contribute to the overall goals and objectives of a business solution a business organization. Research conducted in organizational theory and strategic alliances unfolded several additional agent-based architectural styles that may be used to model the coordination of stakeholders in order to best achieve a given set of goals.

\textsuperscript{10} In simple terms, agents are proactive and reactive (optionally mobile) entities that can act on behalf of users in an autonomous fashion.

\textsuperscript{11} This initial set of process patterns are part of the Requirements Model’s Enterprise Solution Patterns category described in Section \textbf{Error! Reference source not found.}. That category classifies process pattern definitions sub-categories that are maintained by the Project Management Office (PMO). It is part of the responsibilities of the Enterprise Architects to evolve process patterns based on feedback obtained from other architects applying the EAMF methodology to practical projects. Part of this evolution calls for the addition of new process patterns as needed to increase the level of guidance provided by the methodology.
One of the motivations for considering agent patterns when introducing new actors in Actor and Dependency Models may be based on the observation that there is a natural fit for autonomous collaborating entities to take on the responsibilities of one or more actors. Autonomy may be achieved by applying the Agent as Delegate pattern that suggests using agents to act on behalf of a physical actor performing specific tasks. In this case, a business solution agent acting on behalf of a physical actor may manage a set of task agents working on individual tasks, and still has enough autonomy to control access to the physical actor’s data. Delegation implies the existence of a dependency between the physical actor and the agent(s) acting on its behalf. A different problem context where a set of dedicated autonomous collaborating actors need to interact using a common message format may lead to the selection of the Agent as Mediator pattern illustrated in Figure 4. The Agent as Mediator pattern uses a service agent to mediate between the members of a group of agents. In order to interact with each other, agents need to agree on a common message format that is grounded in a common ontology as suggested by the Common Vocabulary illustrated in Figure 16.

As an alternative to agent modeling, collaborating business solution actors may be modeled as loosely coupled location independent business services. In this case, Business Architects would consult the Requirements Model Enterprise Solution Patterns category and would apply the patterns and pattern languages related to the Business Service Society which includes Enterprise Reference Elements such as Component Services, Composite (Business) Services, and Conversational (Workflow) Services. Patterns provided in the Business Service Society focus on describing the interactions between primary business actors in terms of archetypal business interactions such as user-to-business interaction where users access transactions on a 24x7 basis. As a result, these patterns only apply when specific requirements dictate the nature of the interaction and the constraints placed on it. High-level requirements are generally not sufficient to establish a context that motivates the use of the Business Service Society. While the examples of patterns referenced in this section are domain-independent patterns documented in the standard EAMF business pattern catalog, it may be the case that domain-specific patterns from the Enterprise Solution Catalog category are more relevant to the business architecture under study.

As the EAMF methodology gets applied to more and more projects, new standard and domain-specific actor modeling patterns and pattern languages will be explored and documented in the Requirements Model Enterprise Solution Patterns category. In time, the Requirements Model
Enterprise Solution Patterns category will become a priceless knowledge tank of standard, domain-specific, and company specific process patterns that will help architects deliver quality Business Architectures on-time and on-budget. It is the responsibility of Business Architects to understand the relevant EAMF process patterns in order to create and evolve best practice Actors and dependency Models iteratively based on high-level business requirements. Another responsibility of Business Architects is to work closely with Enterprise Architects responsible for evolving the Requirements Model’s Enterprise Solution Patterns category.

**Step 2: Identifying High-Level Tasks**

Figure 3 introduces a *sub-goal* modeling construct that is used as part of another facet of the EAMF reasoning approach described in this section. Figure 3 refers to this second facet of the reasoning approach as step 2 indicated via a circled number “2” on the figure. The sub-goal constructs is used when reasoning about goals and decomposing them into more specific sub-goals or soft-goals. While goals (also referred to as hard-goals) and sub-goals are achievable, soft-goals are not. The purpose of goal decomposition is to refine goals into high-level tasks where a *task* represents a way of satisfying the original goal. Tasks may then be refined further into responsibilities that may be assigned to participants involved in a use case/process. Therefore, refining tasks leads to the elicitation of Business Use Case Requirements that are one step away from associated process requirements. EAMF defers the elicitation of process requirements to the Business Architecture Analysis discipline covered later. In summary, goal decomposition allows the identification of high-level tasks which is a step towards identifying business use case and process requirements.

The type of conceptual model applicable to this second step of the EAMF reasoning approach is referred to as a *Goal and Task Model*. This type of model results from the refinement of goals into high-level tasks. It helps Business Architects analyze goals and tasks from the perspective of a specific actor using various reasoning techniques. Again, these reasoning techniques are based on EAMF process patterns catalogued in the Requirements Model’s Enterprise Solution Patterns category. Possible patterns include means-end analysis, contribution analysis, and AND/OR decomposition. When applied to a given goal, means-end analysis is achieved by refining the goal into sub-goals in order to help identify tasks, resources and soft-goals that provide means to satisfy the goal. Contribution analysis helps Business Architects point out goals that may contribute positively or negatively in reaching the goal being analyzed. Contribution analysis may be considered as a special case of means-end analysis, where means are always goals. Finally, AND/OR decomposition allows refining of goals via combination of AND/OR decompositions from root goals into sub-goals. Business Architects may document multiple Goal and Task Models. However, it is their responsibility and analysis choice to select one of these alternatives to best support the business architecture they are defining.

### 1.1.1.5 Documenting Requirements Modeling Activities

To follow each of the steps described in Sections 1.1.1.3 and 1.1.1.4, Business Architects have to understand why individual artifacts are placed in the Requirements Model. The EAMF methodology recommends documenting these motivations in order to facilitate both the Requirements Model review process and the management of any subsequent changes to requirements provided to Business Architects as analysis input.

---

12 Business Architects may identify multiple alternative Goal and Task models for a given Actor and Dependency Model.
Documenting the reasoning behind the mapping of existing requirements into corresponding requirements and/or definitions within the EAMF requirements model simply amounts to keeping track of traceability relationships between requirements provided as analysis input and requirements and definitions in the Requirements Model.

Documenting the reasoning approach presented in Section 1.1.1.4 amounts to being able to explain the presence of various modeling constructs in Actor and Dependency and Goal and Task Models. To handle this, the EAMF methodology suggests grouping the various modeling constructs used to create these models into three new categories of definitions, namely Elements, Links, and Composite Links. The Elements definition category includes Goal, Soft-goal, Actor, Task, and Resource definitions. The Link definition category includes Dependency, Decomposition, Contribution, and Correlation definitions. Finally, the Composite Link definition category includes And and Or composition definitions. These three new categories of definitions become part of the group of definitions categories contained in the Project Requirements Model Categories. It is then the responsibility of Business Architects to keep track of traceability relationships between definitions in these new categories, as they are being introduced in the models, and the previous group of definitions. Optionally, Business Architects may decide to keep track of association relationships between definitions within these three new categories of definitions. This is not a requirement as these associations should already be documented in the models. However, if they decide to do so, Business Architects will need to use parent-child relationships as needed to host decomposition links and sub-goals definitions under related goal definitions, and to group links definitions under related composite links definitions.

As mentioned earlier, Business Architects may document multiple Actor and Dependency and Goal and Task Models. It is the responsibility of Business Architects to properly categorize the Element, Link, and Composite Link definitions that pertain to these alternatives.

### 1.1.1.6 Populating Requirements Model Categories

As explained in Section 1.1.1.1, the Business Entity Requirements category is used as a placeholder for business entities that may support multiple alternative design choices. Each alternative design choice is supported by a different set of Business Entity Requirements. As Business Architects add new Actor modeling constructs to Actor and Dependency Models, they also document them in the Element Definition category of the Requirements Model, and create corresponding definitions in the Project Business Vocabulary Definition category. These actors are then referenced in the Business Entity Requirements category. Similarly, as Business Architects add new Task modeling constructs to Goal and Task Models, they also document them in the Element Definition category of the Requirements Model. These tasks are then referenced in the Business Entity Requirements category. In addition, as Business Architects refine actor goals, they may identify refined Functional and Non-Functional Requirements. These refined requirements are added to the corresponding Requirements Definition categories in the Requirements Model.

In order to further populate Requirements Model categories, Business Architects must decide between the alternative designs they have documented so far. Their decision consists in picking the alternative that best achieves all the business goals. Once they have selected an alternative design, Business Architects, add Organizational and Location Requirements in the Business Model Requirements category as references to Actor Element definitions in the Requirements Model. Business Architects also add Business Use Case Requirements as references to Task Element definitions in the Requirements Model. As explained in Section 1.1.1.4, the EAMF methodology defers the elicitation of business process model requirements to the Business
1.1.1.7 Building Requirements Models Iteratively

It is generally the case that Business Architects keep evolving the Requirements Model as they refine the original set of requirements further by following the mapping and reasoning steps described in Sections 1.1.1.3 and 1.1.1.4 and continue on with the Business Architecture High-Level Analysis and Design steps described later. Refined requirements may be introduced by Business Architects early or late during the Elaboration phase illustrated in Error! Reference source not found. While the development of Requirement Models is iterative, it is useful for Business Architects to relate to their work on the Requirements Model in terms of “Early Requirements” and “Late Requirements” modeling iterations13. Actual Requirements Model Engineering work conducted by Business Architects during these modeling iterations is as follows:

**Early Requirements Modeling Iterations:**

Relevant participants are identified, along with their respective objectives. Participants are represented as actors, and their objectives are represented as goals. Business Analysts and Business Architects perform this type of Requirements Model Engineering work before Business Architects start working on the Business Architecture in the EAMF business perspective.

**Late Requirements Modeling Iterations:**

The business-solution-to-be is further introduced as an additional (set of) business solution actor(s) and actor dependencies. These dependencies indicate the obligations of the business solution towards its environment. It also indicates what the business solution can expect from actors in its business environment. This type of Requirements Model Engineering work is started by the Business Architect during Requirements Model Engineering and is continued by all Architects as part of the EAMF High-Level Analysis discipline as it applies to all EAMF perspectives.

Actor and Dependency Models are typically built as part of the early requirements modeling iterations. At that time, the initial focus of Business Architects is on characterizing the business domain participants, their intentions, and the dependencies that interleave them. During Requirements Model Engineering, Business Architects may start introducing the business-solution-to-be as another (set of) actors, along with their inter-dependencies with physical actors. Actor and Dependency models are developed further during late requirements modeling iterations by adding more business-solution-to-be actors, along with their inter-dependencies with physical actors.

Goal and Task models are first developed during early requirements modeling iterations using the actors and goals identified initially. Goal and Task models play a similar role in helping identify (and justify) actor dependencies during late requirements modeling iterations.

---

13 As will be explained in Sections Error! Reference source not found., Error! Reference source not found., Error! Reference source not found., Application, Information, Technology, and Enterprise Architects also evolve the Requirements Model. For example, Application Architects typically analyze and evolve Requirements Models as they analyze the application environment and the system-to-be.
1.1.2 EAMF Requirements Model Engineering Implementation

This section details the practical implementation of the EAMF Requirements Engineering methodology presented in the next section in the context of a sample project implementation. The implementation called for the use of several third party tools including Sparx Systems Enterprise Architect (a.k.a., “Sparx Systems EA”), IBM Rational ReqPro, and jUCMNav. IBM Rational ReqPro relies on Microsoft Word to assist Business Analysts and Business Architects with requirements documentation. Requirements can be saved in IBM Rational ReqPro’s underlying repository via macros accessible from the Microsoft Word Main Menu; they can then be manipulated and traced to each other from within the IBM Rational ReqPro client user interface.

Figure 5 illustrates the numbered sequence of activities that Business Analysts followed to assemble Requirements Models as part of the project. As illustrated in the figure, Business Architects collaborated closely with Business Analysts in the early stages of the Requirements Engineering discipline and used high-level requirements created by Business Analysis as an analysis input. Figure 5 emphasizes the various types of artifacts that are created as a result of each activity. It also indicates the specific tools used to create these various artifacts, and the corresponding repository used by each tool. The end result of the sequence of activities shown in Figure 5 is the Requirements Model that will serve as an analysis input to the Business Architecture Analysis discipline.

Details of how the various tools were integrated and used individually to support the sequence activities shown in Figure 5 are provided later on in this section. Section 1.1.2.1 explains how Sparx Systems EA serves as a repository for definitions, references, and traceability information pieced together by Business Architects. Sparx Systems EA also references other artifacts compiled via IBM Rational ReqPro and jUCMNav to support the EAMF disciplines. Section 1.1.2.2 documents and explains the IBM Rational ReqPro project templates and document outlines that were introduced to support the EAMF Requirements Engineering discipline. Section 1.1.2.3 explains the use of jUCMNav to create representative diagrams for the Actor and Dependency and Goal and Task models introduced in Section 1.1.1.4. Finally, Section 1.1.2.4 briefly explains the simplified EAMF catalog infrastructure used to support the project.

---

14 IBM Rational ReqPro uses Microsoft Access as its default database. It is also possible to configure ReqPro to use Microsoft SQL Server, IBM DB2, or Oracle. The default configuration was used for the project mostly due to the project’s aggressive timeline.
Figure 5 – Requirements Model Engineering Activities for the Sample Project

1.1.2.1 Sparx Systems EA Implementation Details

Sparx Systems EA was used to maintain EAMF architectural artifacts for the project. Associated change management tools were not considered due to the project’s aggressive timeline. It should
be noted that the use of Sparx Systems EA was mostly motivated by the fact that some of the project architects had access to the tool and had previous experience with it. In fact a SQL Server repository had already been created to host architectural artifacts from previous projects that did not use the EAMF approach. Another factor that came into play when selecting Sparx Systems EA was the fact that it supports a plug-in for OMG’s Business Process Modeling Notation (BPMN). This was an important consideration since the BPMN notation is used in the EAMF methodology to document Business Process definitions. Project time constraints did not allow for evaluation of alternative tools such as Telelogic Popkin Software Architect or IBM Rational Software Architect. It is our opinion that Sparx Systems EA does quite a good job at bookkeeping architectural artifacts and keeping track of traceability relationships between them. It also supports the creation of templates that are appropriate for use with EAMF, and allows integration of pre-existing requirements specifications documents developed in Word for Windows via IBM Rational ReqPro and/or diagrams developed using jUCMNav.

The initial setup of Sparx Systems EA required a joint effort between Business Analysts and Architects working on the project to come up with generic EAMF-Compliant Project and Enterprise Project Templates. Additional templates were partially created to document the content of the other EAMF-based Enterprise Requirements Models and Pattern/Product/Enterprise Solutions Catalogs. A separate effort, which this document is part of, was initiated to document the EAMF Framework and Methodology. Figure 6 below illustrates the contents of the Sparx Systems EA Requirements Model view as it applies to the Project/EAMF Catalogs/Enterprise Requirements Model Categories introduced in Section 1.1.1.1.

The Sparx Systems EA Requirements Model view is part of the aforementioned EAMF-Compliant Project Template and is used to store Project Requirements Model Categories. Individual template entries are declared in Sparx Systems EA using a stereotyped notation to indicate the nature of the artifacts specified as part of each entry. The template itself is implemented as a root package in Sparx Systems EA that contains several views in addition to the Requirements Model namely the Business, Information, Application, and Technology Architecture views. Each view, at the exception of the Requirements Model view, is stereotyped as a <perspective> to conform to the EAMF grid. The Requirements Model view is labeled as an <analysis input> to indicate that it should initially be used as such by all architects working on a project. Figure 6 below illustrates the template entries used within the Requirements Model view to represent Requirements Model Categories. These entries are stereotyped naturally as a set of <definition> or <requirement> entries to conform to the categories presented in Section 1.1.1.1. In addition they are implemented as individual sub-packages within the Requirements Model view. It is the responsibility of all project architects to evolve the Requirements Model view as they piece together the various facets of a solution architecture for a given business problem.

Figure 6 also illustrates how EAMF Catalogs and Enterprise Requirements Model Categories are labeled in EA. These labels refer to the Pattern Catalog, Product Catalog, Enterprise Solutions Catalog, Enterprise Requirements Model and Enterprise Projects categories. A separate root package called EAMF was created in Sparx Systems EA to host these categories as separate views. These views contain sub-packages used to classify the artifacts typically contained in these categories. For the purpose of the project, the EAMF Catalogs and Enterprise Requirements Model Categories were implemented in simplified form in Sparx Systems EA. It is clear that EAMF artifacts should be stored in their own separate database to support the pattern-driven architecture elicitation approach set forth in EAMF. However, the authors’ main focus during the project was on refining the EAMF methodology and applying it in a practical setting rather than
on building a full scale implementation of the EAMF Framework and Methodology. This explains the limited support implemented to date\textsuperscript{15} to support the EAMF catalogs.

As they start using the Project Requirements provided as analysis input by Business Analysts, Business Architects create a separate IBM Rational ReqPro project using the ReqPro Requirements Model template. Then, they created and populate an IBM Rational ReqPro document referred to as the *Business Architecture Requirements Model Specification*. That document contains all definitions and requirements identified by Business Architects as they closely collaborate with Business Analysts to analyze the Project Requirements. The format of this IBM Rational ReqPro document is discussed in more details in Section 1.1.2.2. As they populate the Business Architecture Requirements Model Specification, which corresponds to step “1” in Figure 5, Business Architects create corresponding `<definition>` and `<requirement>` entries in EA. This applies to definitions contained in the Project Business Vocabulary Definition category and to requirements contained in the Business Use Case Requirements, Business Model Requirements, and Requirements Definition categories. A reference to the Business Architecture Requirements Model Specification is also added at that time in the File Properties of the `<analysis input>`Requirements Model entry.

As they move to Step “2”, as illustrated on Figure 5, Business Architects use the help of Business Analysts to add traceability information between definitions and requirements contained in the Business Architecture Requirements Model Specification document and the various requirements document created by Business Analysts. Once they are done with this step, Business Architects move to the steps labeled “3” and “4” in Figure 5. As indicated in the figure and as documented in Section 1.1.1.4, the end result of these steps is (a set of) Actor and Dependency and Goal and Task diagrams that are representative of the corresponding models. The tool used to create these diagrams is presented in Section 1.1.1.5. For each type of diagram, Business Architects consults the EAMF Enterprise Solution Patterns category to identify process patterns that are applicable to their modeling effort. References to the files that contain these diagrams are added to the Actor and Dependency Models and Goal and Task Models packages located in the `<requirement>`Business Entity Requirements category.

Once they are done with Step “4”, Business Architects move to Step “5” as illustrated in Figure 5. In that step, they document the modeling constructs introduced in the Actor and Dependency and Goal and Task diagrams as per Step “4”. To do so, they add Business Entity requirements to the Business Architecture Requirements Modeling Specification as documented in Section 1.1.1.5. This introduces a Business Entity requirement in Sparx Systems EA for every modeling constructs used in the aforementioned diagrams. In Step “6”, Business Architects add traceability information between modeling constructs and Requirements Model definitions. They do so within EA and within the IBM Rational ReqPro Business Architecture Requirements Modeling Specification. Step “7” follows the approach presented in Section 1.1.1.6 to populate Requirements Model Categories. This results in the addition of `<requirement>` entries in the Business Use Case, Organizational, and Location categories of the Requirements Model. These same requirements are also documented in the Business Architecture Requirements Modeling Specification. Finally, in the last step labeled as Step “8” in Figure 5, Business Architects add

\textsuperscript{15} The same remark applies to the implementation of IBM Rational ReqPro used in the project. This simplified implementation uses Microsoft Access as its underlying database. For scalability, and reliability reason as well as to facilitate true concurrent access, future implementation efforts should plan on deploying an Enterprise repository based on a mainstream DBMS product to handle the storage of requirements maintained via IBM Rational ReqPro.
traceability information in the Business Architecture Requirements Model Specification to document where the requirements identified in Step “7” are derived from.

Figure 6 - Use of Sparx Systems EA to Engineer the EAMF Requirements Model
Figure 7 below is a screenshot of the actual Sparx Systems EA template used in the project. The tools’ project view shown in the figure illustrates the structure of the Requirements Model view explained in the above.
1.1.2.2 IBM Rational ReqPro Implementation Details

IBM Rational RequisitePro (a.k.a., “IBM Rational ReqPro”) was used to handle Requirements Model definition for the project. Associated change management tools were not considered at this time. In order to facilitate the transitioning of information between the Requirements Engineering discipline and the Requirements Model Engineering perspective, it is strongly recommended to use the same tools whenever possible to manage related information. As a result, Business Architects geared up with IBM Rational ReqPro and collaborated with Business Analysts to create generic IBM Rational ReqPro templates applicable to the Project Requirements Model, EAMF Catalogs, Enterprise Requirements Model, and Enterprise Projects Categories illustrated on the right side of Figure 6. The templates devised for the Project Requirements Model and Enterprise Projects categories were modified augmented, as necessary, to include support for the Elements, Links, and Composite Links definitions categories introduced in Section 1.1.1.4.

The template that was created to document the Project Requirements Model category includes IBM Rational ReqPro document outlines for all the document types used to support Requirements Model Engineering in projects. Since definitions and requirement types had already been identified and documented as per Sections 1.1.1.1 and 1.1.1.5, associated prefix were created for them as needed in IBM Rational ReqPro. Various document types were also created as needed to host the corresponding requirements. To facilitate joint work between Business Architects and Business Analysts, the same IBM Rational ReqPro requirements prefixes and document extensions were preserved whenever possible.

In addition to the Project Requirements Model template, an Enterprise Projects template and several EAMF Catalogs and Enterprise Requirements Model templates were partially designed and integrated within IBM Rational ReqPro to support the EAMF Catalogs, Enterprise Requirements Model, and Enterprise Projects categories illustrated on the right side of Figure 6. The template used for the Enterprise Projects category is identical to the aforementioned Project Requirements Model template. The IBM Rational ReqPro templates for the Enterprise Requirements Model Categories were partially created. This applies to the Enterprise Glossary, Enterprise Business Rules, Enterprise Solution Patterns, and Enterprise Strategies definitions and requirements categories. The current implementation of EAMF catalogs relies on the combined use of IBM Rational ReqPro, jUCMNav, and a mainstream RDBMS. Implementation details are covered as a separate topic in Section 1.1.2.4.

In general Enterprise Requirements Models, and Enterprise Projects are defined as Enterprise-level definitions and requirements so they can be re-used across multiple projects by referencing information stored in external IBM Rational ReqPro projects and/or Sparx Systems EA projects. Accordingly, as a result of several observations, some of the activity threads in the project were tied to Enterprise Strategies definitions and requirements that traced back to Innovation requirements set forth as strategies and initiatives by the company’s Strategic Planning Office. Similarly, in addition to including Project-specific definitions and requirements, the ESLM Requirements Model referenced Enterprise Strategies definitions as well as definitions and requirements from the Enterprise Projects Category specific to Enterprise Worker Services Projects. As per Figure 6, it should be noted that the Project Requirements Model Category include Project Business Vocabulary Definitions that are project specific and are not part of the Enterprise Glossary located in the EAMF Enterprise Requirements Model Category.

Figure 8 below illustrates the list of Project and Enterprise Requirements and Documents types and their corresponding IBM Rational ReqPro prefixes and extensions. Requirements types are shown on the left side of the figure and document types are shown on the right side. Business
Architects will need to apply an IBM Rational SODA report template to compile the Requirements Model Specification based on the information contained in the more granular documents catalogued in Figure 8. As illustrated in Figure 8, different IBM Rational ReqPro prefixes may be used for a given Definition or Requirement Type as it is the case for Features and Events (i.e., EVE, and FEA) definitions, or Non Functional Requirements (i.e., requirements with NFE, NFP, and NFR prefixes). Different types of Definitions and Requirements may also be used in a document of a given type as it is the case for Non-Functional Requirements (i.e., requirements with NFE, NFP, and NFR prefixes) within High Level/Detailed Requirements Definition documents (i.e., documents with .HRD or .DRD extensions). Note that Requirements Model Specification documents typically contain many requirements of different types.

**Figure 8 - Use of IBM Rational ReqPro for the Requirements Model Engineering Phase**

Figure 9 below is a screenshot of a sample IBM Rational ReqPro project that uses the Project Requirements Model template described in this section. The IBM Rational ReqPro project explorer shown on the left pane lists some of the project Functional Requirements for Service
Request Processing. The right pane illustrates an IBM Rational ReqPro traceability matrix view that illustrates traceability between Functional Requirements and Business Objectives in the Project Requirements and related requirements in the Project Requirements Model provided as analysis input by Business Analysts. At all times, Business Architects working with definitions and requirements artifacts in Sparx Systems EA have access to the traceability of these artifacts all the way back to the original requirements created by Business Analysts. This traceability will help guarantee that Architects are building the “right product” (i.e., the product that meets the requirements) rather than simply building the product right. The EAMF methodology described in this document carries this traceability forward to other EAMF perspectives so that all architectural artifacts maintained in Sparx Systems EA may be traced back to requirements captured in IBM Rational ReqPro.

Figure 9 - RequisitePro Project Requirements Model Template for the Project

1.1.2.3 Additional EAMF Implementation Tools

The focus of EAMF is to provide an end-to-end modeling approach that starts all the way from high-level business requirements and delivers best practice solution architectures that can be validated and readily implemented. Sparx Systems EA does not support the Actor/Dependency and Goal/Plan modeling described in Section 1.1.1.4. The jUCMNav tool was used to develop the Actor/Dependency and Goal/Plan models for ESML as part of the EAMF Requirements Model Engineering activity. jUCMNav is a general, goal-oriented and/or agent-oriented modeling and analysis tool. It provides users with a graphical interface that helps develop models by leveraging knowledge and allowing sophisticated computer-aided analysis.

jUCMNav supports the User Requirements Notation [BPMURN], and is compatible with the [GRL], and IStar [SARMI*] modeling notations. URN is an all-encompassing notation used by EAMF to help analyze and design architectural solutions. URN transcends goal-oriented/use-case
driven design, Service Oriented Analysis and Design (SOAD), and other high-level modeling guidelines provided by Enterprise Application Frameworks (EAFs). EAMF leverages the fact that URN and UML notations complement each other to address key differences\textsuperscript{16} between object-oriented and goal/agent-oriented development methodologies. To take advantage of the complementary nature of the URN and UML notations, EAMF relies on the jUCMNav tool, which can interoperate with UML-based modeling tools. The GRL [GRL] notation was used to draw Actor/Dependency and Goal/Task diagrams in jUCMNav. The GRL notations support the modeling constructs presented in Section 1.1.1.4 as illustrated in Figure 10. The jUCMNav tool is available as an Eclipse plug-in. Version 2.0.1 of that plug-in was used to support the project. jUCMNav can save its models in .jucm file format and .xml format. For the project, the models were saved in .jucm file format and were referenced as documents within the Sparx Systems EA project documentation template. Corresponding references are located under the Business Entity Requirements package.

While tools like IBM Rational ReqPro assist Business Analysts with requirements classification and traceability, they fall short of providing a methodology for reasoning about requirements when the business environment and the business-solution-to-be are analyzed. This kind of requirements modeling is different from the detailed specification of what is to be done. Here the modeler is primarily concerned with exposing "why" certain choices for behavior and/or structure are made or constraints are introduced. The modeler is not yet interested in the "operational" details of processes or system requirements. The integration of explicit goal representations in the requirements model provides a criterion for requirements completeness. Requirements may be considered complete if they are sufficient to establish the goal they are refining.

GRL modeling constructs available within jUCMNav are illustrated in Figure 10. When using the GRL notation within jUCMNav. Actors and Dependency models are graphically represented through Actor and Dependency diagrams in which actors are depicted as circles, their goals/sub-goals as ovals and their soft-goals as cloud shapes. The network of dependency relationships among actors are depicted using various types of links depicted as arrowed lines. Goal and Task models are depicted through Goal and Task diagrams that represent the perspective of a specific actor as a balloon that contains graphs whose nodes are goals (ovals) and/or tasks (hexagonal shape) and whose arcs represent the different types of relationships that can be identified between nodes. Soft-goal analysis may be conducted within an actor balloon to identify positive or negative contributions from other goals.

\textsuperscript{16} Agents (and actor/dependency diagrams) cannot be thought as a specialization of UML objects (and class diagrams). There are distinct representational and ontological levels used for class diagrams and actor diagrams (the former being at the software level, the latter at the knowledge level). Moreover, inheritance, a crucial concept for UML diagram, plays no role in actor/dependency diagrams. Similar differences exist between goal/task diagrams and UML notations.
Figure 10 - jUCMNav Modeling Constructs [DEBMU]

A screenshot of the jUCMNav user interface is illustrated in Figure 11 below. The figure illustrates the editing of a sample GRL diagram within jUCMNav. The GRL modeling constructs available within jUCMNav are shown in the palette view located on the far right side.
1.1.2.4 EAMF Catalog Database

As illustrated in Figure 6, Enterprise-wide EAMF Catalogs supported in IBM Rational ReqPro include Pattern, Product, and Enterprise Solutions Catalogs. The implementation of these catalogs currently relies on a combination of IBM Rational ReqPro, jUCMNav, and a mainstream RDBMS. The role of the pattern and Enterprise Solution catalog is to facilitate the book-keeping of standard and domain-specific pattern languages that applies to various solution domains. Individual patterns in a given pattern language are represented/described in IBM Rational ReqPro documents. These documents contain pattern specific information such as the name, high-level description, problem description, solution, and pattern solution Force Hierarchy model. Force hierarchy models are documented using GRL-based Force Hierarchy diagrams. These diagrams are created using the GRL notation supported by jUCMNav.

It is the responsibility of Business Architects to leverage the EAMF pattern catalog in order to create pattern languages for given problem domains. A pattern language for a problem domain is created by capturing the force hierarchy that best describes the problem. This information is typically captured in a GRL-based Problem Domain Force Hierarchy diagram. That force hierarchy diagram is then used to identify the solution domain that most closely matches the problem. This in turn provides a set of patterns and their corresponding force hierarchies.
Individual patterns are then selected based on how well their force hierarchy addresses functional and non-functional characteristics of the problem domain under study\(^{17}\). The various pattern languages created by project and Enterprise architects working in different EAMF perspectives are represented as Pattern Cluster Networks in the EAMF grid. The elicitation of Pattern Cluster Networks applicable to Business Architectures is documented later.

While the Pattern and Enterprise Solution Catalogs are used to support architectural modeling as part of the analysis and design disciplines, the Product Catalog helps support the selection of Enterprise-approved third party products that may be used to implement the characteristics of pattern languages that Business Architects have identified as solutions for given problem domains. The elicitation of Product Cluster Networks applicable to Business Architectures is documented later.

### 1.2 Enterprise Service Lifecycle Management (ESLM)

#### Requirements Engineering and EAMF Requirements Model Engineering Sample Project

**1.2.1 ESLM EAMF Requirements Model Engineering**

Business Architects followed the steps outlined in Sections 1.1.1.3-1.1.1.6 to put together a Requirements Model for the ESLM project. As a preliminary step, they used the Project Requirements Model template in IBM Rational ReqPro to create a project named “ESLM-RM”. They also created an “Enterprise Service Lifecycle Management” project in the Sparx Systems EA Repository using the “EAMF Base Project” as a template. The Requirements Model was built iteratively as described in Section 1.1.1.7 as part of Early Requirements iterations based on the EAMF Requirements Model Engineering discipline.

**1.2.1.1 Mapping Requirements into the EAMF Requirements Model**

The following describes the sequence of steps that was followed by Business Architects to identify and document definitions and requirements for the ESLM Requirements Model. Corresponding activities are labeled as “1” and “2” in Figure 5. Practically, the steps described in the following involved the following:

- Creation of document-based\(^ {18}\) definitions and requirements in the “ESLM-RM” IBM Rational ReqPro project using the various document types illustrated in Figure 8. At this “early requirement” stage, detailed use case, detailed business model, and detailed requirements definition documents were not considered.

---

\(^{17}\) The elicitation of a solution domain that most closely matches a problem, and the selection of corresponding patterns that best address the functional and non-functional characteristics of a given problem domain relies on support provided by the underlying EAMF infrastructure. Manual elicitation is clearly overwhelming and it is the goal of EAMF to provide various algorithms to semi-automate these tedious tasks. Possible algorithms include straight matching, match mining, and rule-based inferring. The exact nature of these algorithms is out of the scope of this document which mostly focuses on the EAMF methodology.

\(^{18}\) It is important to note that all IBM Rational ReqPro Requirements Model definitions and requirements are always document-based to facilitate the generation of specifications that can be reviewed and signed off upon. IBM Rational ReqPro database-only requirements are used in this section to create placeholders that facilitate the viewing of external requirements (e.g., TERM1, SRQ1, etc.)
Creation of corresponding <definition> and <requirement> entries under the <analysis input>Requirements Model view in the Sparx Systems EA “Enterprise Service Lifecycle Management” project

Documentation of traceability between definitions and requirements within the Sparx Systems EA “Enterprise Service Lifecycle Management” project

Documentation of traceability between definitions and requirements contained in the “ESLM-RM” IBM Rational ReqPro project and requirements contained in the “ESLM” IBM Rational ReqPro project.

In general, when it comes to documenting traceability within or across IBM Rational ReqPro projects, it is recommended to do so once all definitions and requirements have been identified and documented. This limits the amount of rework that Business Architects need to put in to clear suspect marks automatically created by IBM Rational ReqPro as inter-related definitions and requirements are being reworked.

**Step 1: Gather Existing Requirements Documentation**

In this step, Business Architects obtained the High-Level Project Requirements (including all traceability information) developed by Business Analysis. This information was available in electronic form via the “ESLM” ReqPro project created by Business Analysts.

**Step 2: Mapping Glossary Requirements into Project Business Vocabulary Definitions**

The “ESLM-RM” IBM Rational ReqPro project stores Project Business Vocabulary definitions under the “Project Business Vocabulary Definition” package. Within that package, a special requirement referred to as “TERM1” is used to facilitate the viewing of external requirements. In addition a view is provided to trace “Project Business vocabulary To/From External References”. Using this view, Business Architects traced TERM1 to existing Glossary requirements previously identified by Business Analysts to indicate which Glossary requirements should be used as Project Business Vocabulary definitions within the “ESLM-RM” IBM Rational ReqPro project. This scheme allows Business Architects to decide as to the applicability of a Glossary requirement and does not require duplication of existing requirements data.

The Sparx Systems EA “Enterprise Service Lifecycle Management” project stores Project Business Vocabulary definitions in the <definition>Project Business Vocabulary package under the <analysis input>Requirements Model view. Business architects added a document reference under that view for their project-specific Project Business Vocabulary Definitions document. In general, Business Architects should add document references for all IBM Rational ReqPro documents known to maintain Project Business Vocabulary definitions for the Requirements Model. This typically includes Project Business Vocabulary Definition documents, and in some cases Enterprise Glossary documents.

In addition to creating Project Business Vocabulary definitions as references to pre-existing Glossary requirements, Business Architects created additional Project Business Vocabulary definitions during subsequent iterations. This happened in particular as they reasoned about business entities’ dependencies and goals as documented in Section 1.2.1.2. For traceability reasons, Business Architects had to document these additional Project Business Vocabulary entries in both the “ESLM-RM” IBM Rational ReqPro project and the Sparx Systems EA “Enterprise Service Lifecycle Management” project. In general, in order to add new Project Business Vocabulary definitions, Business Architects first need to create a document of type “Project Business Vocabulary Definition Document” within the “ESLM-RM” IBM Rational
ReqPro project. New Project Business Vocabulary definitions may then be inserted in this new document. Business Architects may also decide to add definitions to the Enterprise Glossary. In this case, these definitions will need to be referenced via a view within the “ESLM-RM” Project Requirements Model using a scheme similar to that described above in the first paragraph of this step. The current implementation of the IBM Rational ReqPro Project Requirements Model template does not include views to reference definitions and requirements located in Enterprise Requirements Model Categories. Therefore, project-specific private views must be created by Business Architects as needed.

As new Project Business Vocabulary definitions were added to the “ESLM-RM” Project Requirements Model, Business Architects also added them to the Sparx Systems EA “Enterprise Service Lifecycle Management” project in the <definition>Project Business Vocabulary package under the <analysis input>Requirements Model view.

This general approach at documenting Project Business Vocabulary Definitions allows a clean separation between requirements engineering activities and requirements model engineering activities. At the same time, it guarantees full traceability of definitions derived as part of the requirements model discipline all the way back to requirements created by Business Analysts as part of the requirements engineering discipline. The same general approach applies to the documentation of the other Requirements Model definitions and requirements covered in this section.

**Step 3: Mapping Stakeholder Requirements into Strategy Definitions**

The “ESLM-RM” IBM Rational ReqPro project stores Strategy definitions under the “Strategy Definition” package. Within that package, a special requirement referred to as “SRQ1” is used to facilitate the viewing of external requirements. In addition a view is provided to trace “Strategy Definitions To/From External References”. Using this view, Business Architects traced SRQ1 to existing Stakeholder Requests requirements previously identified by Business Analysts to indicate which Stakeholder Requests should be used as Strategy Definitions within the “ESLM-RM” IBM Rational ReqPro project. This scheme allows Business Architects to decide as to the applicability of a Stakeholder Request and does not require duplication of existing requirements data.

The Sparx Systems EA “Enterprise Service Lifecycle Management” project stores Strategy definitions in the <definition>Strategy Definition package under the <analysis input>Requirements Model view. Business architects added a document reference under that view for their project-specific Strategy Definitions document. In general, Business Architects should add document references for all IBM Rational ReqPro documents known to maintain Strategy definitions for the Requirements Model. This typically includes Strategy Definition documents, and in some cases Enterprise Strategies documents.

**Step 4: Mapping Existing Related Requirements into Concept Definitions**

The “ESLM-RM” IBM Rational ReqPro project stores Concept definitions under the “Concept Definition” package. That package contains three sub-packages including “Business Objectives”, “Project Business Directives”, and “Features and Events”.

Within the Business Objectives package, a special requirement referred to as “BUS1” is used to facilitate the viewing of external requirements. In addition a view is provided to trace “Business Objectives To/From External References”. Using this view, Business Architects traced BUS1 to
existing Business Objectives requirements previously identified by Business Analysts to indicate which Business Objectives should be used as Business Objectives definitions within the “ESLM-RM” IBM Rational ReqPro project. This scheme allows Business Architects to decide as to the applicability of a Business Objectives requirement and does not require duplication of existing requirements data.

Within the Project Business Directives package, a special requirement referred to as “BUD1” is used to facilitate the viewing of external requirements. In addition a view is provided to trace “Business Directives To/From External References”. Using this view, Business Architects traced BUD1 to existing Business Policies requirements previously created by Business Analysts in the Business Rules Requirements Repository to indicate which Business Policies should be used as Business Directives definitions within the “ESLM-RM” IBM Rational ReqPro project. This scheme allows Business Architects to decide as to the applicability of a Business Policies requirement and does not require duplication of existing requirements data.

Within the Features and Events package, special requirements referred to as “EVE1” and “FEA1” are used to facilitate the viewing of external requirements. In addition corresponding views are provided to trace “Events To/From External References” and “Features To/From External References”. Using these views, Business Architects respectively traced EVE1 and FEA1 to existing Events and Features requirements previously identified by Business Analysts to indicate which Events and Features requirements should be used as Events and Features definitions within the “ESLM-RM” IBM Rational ReqPro project. This scheme allows Business Architects to decide as to the applicability of Events and Features requirements and does not require duplication of existing requirements data.

The Sparx Systems EA “Enterprise Service Lifecycle Management” project stores Concept definitions in the <definition>Concept Definition package under the <analysis input>Requirements Model view. Business architects added a document reference under that view for their project-specific Concept Definitions document. In general, Business Architects should add document references for all IBM Rational ReqPro documents known to maintain Concept definitions for the Requirements Model. This typically includes Concept Definition documents, and in some cases Enterprise Business Rules documents.

In addition to creating Concept definitions as references to pre-existing Business Objectives, Business Policies, and Features and Events requirements, Business Architects created additional Concept definitions during subsequent iterations. This happened in particular as they reasoned about business entities’ dependencies and goals as documented in Section 1.2.1.2. For traceability reasons, Business Architects had to document these additional Concept entries in both the “ESLM-RM” IBM Rational ReqPro project and the Sparx Systems EA “Enterprise Service Lifecycle Management” project. In general, in order to add new Concept definitions, Business Architects first need to create a document of type “Concept Definition Document” within the “ESLM-RM” IBM Rational ReqPro project. New Concept definitions may then be inserted in this new document. Business Architects may also decide to add definitions to the Enterprise Business Rules. In this case, these definitions will need to be referenced via a view within the “ESLM-RM” Project Requirements Model using a scheme similar to that described above in the first paragraph of this step. The current implementation of the IBM Rational ReqPro Project Requirements Model template does not include views to reference definitions and requirements located in Enterprise Requirements Model Categories. Therefore, project-specific private views must be created by Business Architects as needed.
As new Concept definitions were added to the “ESLM-RM” Project Requirements Model, Business Architects also added them to the Sparx Systems EA “Enterprise Service Lifecycle Management” project in the <definition>Concept Definition package under the <analysis input>Requirements Model view.

**Step 5: Mapping Existing Related Requirements into Functional and Non-Functional Requirements**

The “ESLM-RM” IBM Rational ReqPro project stores Requirements definitions under the “Requirements Definitions” package. That package contains two sub-packages including “Functional Requirements”, and “Non-Functional Requirements”.

Within the Functional Requirements package, a special requirement referred to as “FNC1” is used to facilitate the viewing of external requirements. In addition a view is provided to trace “Functional Requirements To/From External References”. Using this view, Business Architects traced FNC1 to existing Business or Functional requirements previously identified by Business Analysts to indicate which Business or Functional requirements should be used as Functional Requirements definitions within the “ESLM-RM” IBM Rational ReqPro project. This scheme allows Business Architects to decide as to the applicability of a Business or Functional requirement and does not require duplication of existing requirements data.

The Non-Functional Requirements package contains three sub-packages including “Non-Functional External Requirements”, “Non-Functional Process Requirements”, and “Non-Functional Project Requirements”. Within these three sub-packages, special requirements referred to respectively as “NFE1”, “NFP1”, and “NFR1” are used to facilitate the viewing of external requirements. In addition corresponding views are provided respectively to trace “Non-Functional External To/From External References”, “Non-Functional Process To/From External References”, and “Non-Functional Project To/From External References”. Using these views, Business Architects respectively traced NFE1, NFP1, and NFR1 to existing Non-Functional requirements previously identified by Business Analysts to indicate which Non-Functional requirements previously identified by Business Analysts should be used as Non-Functional Requirements Definitions within the “ESLM-RM” IBM Rational ReqPro project. This scheme allows Business Architects to decide as to the applicability of Non-Functional requirements and does not require duplication of existing requirements data.

The Sparx Systems EA “Enterprise Service Lifecycle Management” project stores Requirements definitions in the <requirement>Requirements Definitions package under the <analysis input>Requirements Model view. Business Architects added a document reference under that view for their project-specific High-Level Requirements Definitions document.

In addition to creating Requirements definitions as references to pre-existing Business or Functional Requirements and Non-Functional Requirements, Business Architects created additional Requirements Definitions during subsequent iterations. This happened in particular as they reasoned about business entities’ dependencies and goals as documented in Section 1.2.1.2 and as they explored Business Model Requirements alternatives. For traceability reasons, Business Architects had to document these additional Requirements entries in both the “ESLM-RM” IBM Rational ReqPro project and the Sparx Systems EA “Enterprise Service Lifecycle Management” project. In general, in order to add new Requirements definitions, Business Architects first need to create a document of type “High-Level Requirements Definition Document” or “Detailed Requirements Definition Document” within the “ESLM-RM” IBM
Rational ReqPro project. New Requirements definitions may then be inserted in this new document.

As new Requirements definitions were added to the “ESLM-RM” Project Requirements Model, Business Architects also added them to the Sparx Systems EA “Enterprise Service Lifecycle Management” project in the <requirement>Requirements Definitions package under the <analysis input>Requirements Model view.

Step 6: Mapping Existing Use Case Requirements into Business Use Case Requirements

The “ESLM-RM” IBM Rational ReqPro project stores Business Use Case requirements under the “Business Use Case Requirements” package. Within that package, a special requirement referred to as “USE1” is used to facilitate the viewing of external requirements. In addition a view is provided to trace “Business Use Cases To/From External References”. Using this view, Business Architects traced USE1 to existing Use Cases requirements previously identified by Business Analysts to indicate which Use Cases should be used as Business Use Cases Requirements within the “ESLM-RM” IBM Rational ReqPro project. This scheme allows Business Architects to decide as to the applicability of a Use Cases requirement and does not require duplication of existing requirements data.


In addition to creating Business Use Cases requirements as references to pre-existing Use Cases requirements, Business Architects created additional Business Use Cases requirements during subsequent iterations. This happened in particular as they reasoned about business entities’ dependencies and goals as documented in Section 1.2.1.2. For traceability reasons, Business Architects had to document these additional Business Use Cases entries in both the “ESLM-RM” IBM Rational ReqPro project and the Sparx Systems EA “Enterprise Service Lifecycle Management” project. In general, in order to add new Business Use Cases requirements, Business Architects first need to create a document of type “High-Level Business Use Cases Document” or “Detailed Business Use Cases Document” within the “ESLM-RM” IBM Rational ReqPro project. New Business Use Cases requirements may then be inserted in this new document.

As new Business Use Cases requirements were added to the “ESLM-RM” Project Requirements Model, Business Architects also added them to the Sparx Systems EA “Enterprise Service Lifecycle Management” project in the <requirement>Business Use Cases Requirements package under the <analysis input>Requirements Model view.

Step 7: Mapping Existing Related Requirements into Business Model Requirements

The “ESLM-RM” IBM Rational ReqPro project stores Business Model requirements under the “Business Model Requirements” package. That package contains three sub-packages including “Location Requirements”, “Organizational Requirements”, and “Process Model Requirements”.

Within the Location Requirements package, a special requirement referred to as “LOC1” is used to facilitate the viewing of external requirements. In addition a view is provided to trace “Location To/From External References”. Using this view, Business Architects traced LOC1 to existing Location requirements previously identified by Business Analysts to indicate which
Location requirements previously identified by Business Analysts should be used as Location Requirements within the “ESLM-RM” IBM Rational ReqPro project. This scheme allows Business Architects to decide as to the applicability of a Location requirement and does not require duplication of existing requirements data.

Within the Organizational Requirements package, a special requirement referred to as “ACT1” is used to facilitate the viewing of external requirements. In addition a view is provided to trace “Organizational To/From External References”. Using this view, Business Architects traced ACT1 to existing Organization requirements previously identified by Business Analysts to indicate which Organization requirements previously identified by Business Analysts should be used as Organizational requirements within the “ESLM-RM” IBM Rational ReqPro project. This scheme allows Business Architects to decide as to the applicability of an Organization requirement and does not require duplication of existing requirements data.

The Process Model Requirements package contains three sub-packages including “Business Process Requirements”, “Business Rules Requirements”, and “Workflow Rules Requirements”. Within these three sub-packages, special requirements referred to respectively as “BPR1”, “BUR1”, and “WFR1” are used to facilitate the viewing of external requirements. In addition corresponding views are provided respectively to trace “Business Process To/From External References”, “Business Rules To/From External References”, and “Workflow Rules To/From External References”. Using these views, Business Architects respectively traced BPR1, BUR1, and WFR1 to existing Business Model requirements previously identified by Business Analysts to indicate which Business Model requirements previously identified by Business Analysts should be used as Business Model Requirements within the “ESLM-RM” IBM Rational ReqPro project. This scheme allows Business Architects to decide as to the applicability of Business Model requirements and does not require duplication of existing requirements data.


In addition to creating Business Model requirements as references to pre-existing Business Model requirements, Business Architects created additional Business Model requirements during subsequent iterations. This happened in particular as they reasoned about business entities’ dependencies and goals as documented in Section 1.2.1.2. For traceability reasons, Business Architects had to document these additional Business Model entries in both the “ESLM-RM” IBM Rational ReqPro project and the Sparx Systems EA “Enterprise Service Lifecycle Management” project. In general, in order to add new Business Model requirements, Business Architects first need to create a document of type “High-Level Business Model Document” or “Detailed Business Model Document” within the “ESLM-RM” IBM Rational ReqPro project. New Business Model requirements may then be inserted in this new document.

As new Business Model requirements were added to the “ESLM-RM” Project Requirements Model, Business Architects also added them to the Sparx Systems EA “Enterprise Service Lifecycle Management” project in the <requirement>Business Model Requirements package under the <analysis input>Requirements Model view.
1.2.1.2 Reasoning about Business Entities’ Dependencies and Goals

The following describes the sequence of steps that was followed by Business Architects to elicit and document new requirements in the ESLM Requirements Model as a result of applying the approach described in Section 1.1.1.4 and documented in Section 1.1.1.6. Corresponding activities are labeled as “3” and “4” in Figure 5. Practically, the steps described in the following involved the creation of actor and dependency and goal and task diagrams using the jUCMNav tool. On one hand, actor and dependency diagrams provide a way to analyze requirements and identify business entities and their relationships. On the other hand, goal and task analysis diagrams help establish requirements dynamics.

Step 1: Capturing Participants Requirements and Flow Requirements Between Them

A cursory inspection of the problem domain requirements identifies two actors initially referred to as the Service Requestor and Service Request Status Tracking Actor. The Service Requestor may be a physical agent of a business solution agent. The Service Request Status Tracking Actor is an autonomous entity in the sense that it does not require the Service Requestor’s approval at every step of executing its task. In other words, it can act on its own. As a result, the Service Request Status Tracking Actor is renamed to Service Request Status Tracking Agent, and its autonomous characteristic motivates the selection of an agent pattern to model it. Modeling the Service Request Status Tracking Actor at this stage must conform to existing requirements. A review of the global forces for the Agent pattern roadmap and the specific context and forces listed for the Agent as Delegate pattern in [PLMUA] suggest a pattern match. The Service Request Status Tracking Agent must remain autonomous to keep interacting with Service Requestor(s) and ensure “time efficiency of the interaction”. It must also emphasize other aspects such as “security of information exchanged”. Finally, it must ensure the “accuracy of status tracking information”. To meet the global force of autonomy and the additional problem domain forces listed, the best approach for the Service Request Status Tracking Agent is to act as a User Agent that delegates time-consuming and/or peripheral tasks to one or more Task Agent(s) as per the Agent as Delegate pattern illustrated in Figure 4. The time-consuming task in this case is the service request itself, and it therefore allocated to a new autonomous agent called the Service Request Processor.

Figure 12 shows the actor and dependency diagram for the ESLM domain as resulting from a first early requirement analysis. Actors and goal naming conventions must be followed closely and should comply with the terminology defined in the project business vocabulary. The actors of interest at this stage are major business agents whose intervention is required to meet the business objectives listed in the previous section. Glossary requirements can be used to help identify and name these major agents. Business agents’ objectives are considered as goals. In particular, the generic actor Service Request Status Tracking Agent has one main goal: capture and store service request information and one subgoal: provide accurate service request status information. The actor Service Requestor has two associated goals initiate service request on a business entity and consume service request status information. The actor Service Request Processor wants to store and retrieve service request status information.

---

19 It is the responsibility of Enterprise Architects to document relevant pattern societies and associated pattern languages in the EAMF pattern catalog. Due to the nascent state of the EAMF catalog at this time and the aggressive timeline on the project, the authors had to research relevant patterns in the literature and did not get a chance to document them in the catalog. Properly documenting patterns in the EAMF catalog will cut down research time and provide a uniform documentation for the different patterns that will facilitate the selection of patterns and speed up the analysis process.
Step 2: Identifying High-Level Tasks

As documented in Error! Reference source not found., the main business objective of the ESLM project is to devise a solution which is able to locate any type of service transaction that enters the company and enable its processing by one or more business processes and/or workflows with 100% accuracy. The ESLM requirements specification (i.e., Error! Reference source not found.) identifies business model, domain model, and functional Business requirements. It is important to note that the RequisitePro project used in this document is the one that was produced to describe the requirements of ETST (and then ESLM) rather than the original project. The evolution of the requirements documents within RequisitePro is in line with the iterative approach subsumed in the methodology described in this document.

1.2.1.3 Documenting Requirements Modeling Activity

Corresponding activities are labeled as “5” and “6” in Figure 5
1.2.1.4 Populating Requirements Model Categories

Corresponding activities are labeled as “7” and “8” in Figure 5
<table>
<thead>
<tr>
<th>Term</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Actor</td>
<td>An entity that has goals and intentions. An actor represents a physical Agent (e.g., a person, an animal, a car), or a business solution agent as well as a role or a position. A role characterizes the behavior of an actor within a particular context, and a position represents a set of roles. An agent can occupy a position, and a position is said to cover a role.</td>
</tr>
<tr>
<td>Agent</td>
<td>While, there is no universally accepted definition of the notion of agent, the following four properties are generally used to characterize agents: autonomy, social ability, reactivity, and proactiveness. Agents are autonomous computational entities in the sense that they may operate without supervision. Their reactivity is tied to their ability to interact with their environment. They have social abilities that make that they are also able to interact with other agents. Finally, agents are proactive in achieving their own goals. Agents typically represent users and act on their behalf. Therefore, they are treated as actors in much the same way humans are and they provide a metaphor for conceptualizing business and/or software solutions.</td>
</tr>
<tr>
<td>Business Model</td>
<td>A set of participants and the flows between them. The participants include the company behind the business model of interest and all the actors involved. Value is created in the form of information, product, and money flows between the participants.</td>
</tr>
<tr>
<td>Business Scenario</td>
<td>A business scenario is a sequence of responsibilities that must be executed by a business solution and the users of the business solution to achieve a given business functionality.</td>
</tr>
<tr>
<td>Capability</td>
<td>A capability represents the ability of an actor to define, choose and execute a task to satisfy a goal within a particular business environment.</td>
</tr>
<tr>
<td>Dependency</td>
<td>A dependency between two actors indicates that one actor depends on another in order to satisfy a goal, execute a task, or deliver a resource. Via dependencies on other actors, an actor is usually able to satisfy goals that it would otherwise be unable to achieve on its own, or not as easily, or not as well. These dependencies may also have an adverse effect on the ability of an actor to satisfy a goal if some of the actors that are relied on failed to achieve the sub-goals that are delegated to them.</td>
</tr>
<tr>
<td>Force Hierarchy</td>
<td>A hierarchical graph of forces and their contributions on each other, where forces are expressed as a combination of a small set of constraints on common resources such as computation, effort, money, space, time, etc.</td>
</tr>
<tr>
<td>Goal</td>
<td>A goal represents the strategic interests of an actor. Goals are often referred to as hard goals which differ from soft-goals. While goals are achievable, there are no criteria as to whether or not soft-goals may be satisfied. Soft-goals are typically used to model business solution qualities such as security, reliability, performance, and maintainability. Goals may be decomposed and refined into sub-goals.</td>
</tr>
</tbody>
</table>
| Pattern            | Patterns are reusable solutions to recurring design problems. They provide a vocabulary for communicating these solutions to others. However, the documentation of a pattern goes beyond documenting a problem and its
solution. It also describes the forces or design constraints that give rise to
the proposed solution. These undocumented design features are often
described as the “intent” leading to the use of a pattern. Forces can be
thought of as pushing or pulling the problem towards different solutions. A
good pattern balances the forces.

<table>
<thead>
<tr>
<th><strong>Pattern Language</strong></th>
<th>A set of patterns that build on each other.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Process Pattern</strong></td>
<td>In this document, a process pattern is a pattern which describes a proven, successful approach and/or series of actions for developing software.</td>
</tr>
<tr>
<td><strong>Resource</strong></td>
<td>A resource represents the ability of an actor with the ability to define, choose, and execute a task to satisfy a goal within a particular business environment.</td>
</tr>
<tr>
<td><strong>Scenario</strong></td>
<td>A scenario is an instance of a use case, and represents a single path through the use case. Thus, one may construct a scenario for the main flow through a use case, and other scenarios for each possible variation of flow through the use case (e.g., triggered by options, error conditions, security breaches, etc.). Scenarios may be depicted using UCM diagrams, Message Sequence Chars (MSCs) or UML sequence diagrams at different stages of the SDLC.</td>
</tr>
<tr>
<td><strong>Storyboard</strong></td>
<td>In the context of software architectures, a storyboard is an annotated diagram used to illustrate and explain a business scenario.</td>
</tr>
<tr>
<td><strong>Task</strong></td>
<td>A task represents a way of satisfying a goal.</td>
</tr>
<tr>
<td><strong>Attribute-Based Architectural Styles (ABASs) - OLD</strong></td>
<td>ABASs build on architectural styles to provide a foundation for more precise reasoning about architectural design by explicitly associating a reasoning framework (whether qualitative or quantitative) with an architectural style. These reasoning frameworks (e.g., URN, NFR framework, etc.) are based on quality attribute-specific models, which exist in the various quality attribute communities (such as the performance and reliability communities).</td>
</tr>
</tbody>
</table>
## References

<table>
<thead>
<tr>
<th>Information</th>
<th>Source</th>
</tr>
</thead>
<tbody>
<tr>
<td>jUCMNav</td>
<td><a href="http://www.softwareengineering.ca/jucmnav/">http://www.softwareengineering.ca/jucmnav/</a></td>
</tr>
<tr>
<td>[SARMI*]</td>
<td><a href="http://www.cs.utoronto.ca/pub/eric/tut1.1-v2.ppt#256,1,Strategic">http://www.cs.utoronto.ca/pub/eric/tut1.1-v2.ppt#256,1,Strategic</a> Actor Relationships Modelling with <em>i</em></td>
</tr>
<tr>
<td>Library Ref</td>
<td><a href="http://jucmnav.softwareengineering.ca/twiki/bin/view/UCM/UCMVirtualLibrary">http://jucmnav.softwareengineering.ca/twiki/bin/view/UCM/UCMVirtualLibrary</a></td>
</tr>
</tbody>
</table>