D. Volere Requirements Knowledge Model

The Volere Requirements Knowledge Model (**Figure D.1**) provides a language for the knowledge that you discover and accumulate during your requirements activities. We present it as a guide to the information you need to discover, and as a tool for communication between the various stakeholders on your project. The model can also serve as your specification for which requirements knowledge you plan to discover and trace. Your own process must define who gathers which information, to what degree of detail, and how it will be packaged and reviewed.

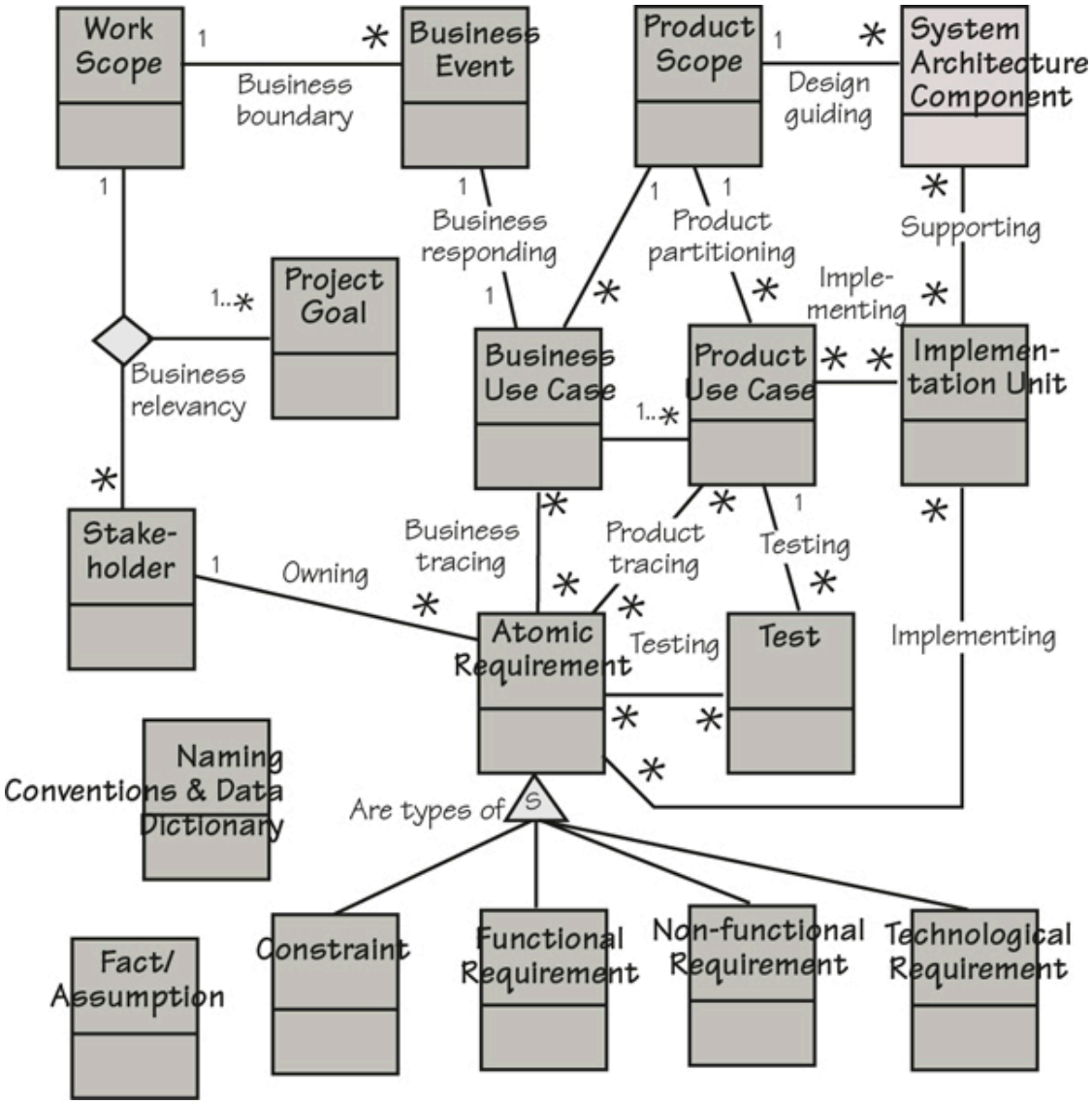

Figure D.1. The knowledge model identifies the classes of knowledge concerned with requirements and the associations between them.

If you are not familiar with this notation:

• A rectangle represents a *class* of knowledge. The name of the class is written in the rectangle.

• A line represents an *association* between two or more classes.

• *Multiplicity* is shown by 1 and *. This means if you have only one of a class, there are * (many) of the class at the other end of the association.

Definitions of Requirements Knowledge Classes and Associations

The purpose of the knowledge model is to provide you with a common of language for communicating and managing requirements. Use the model

as a starting point and then, if you need to, add other classes and associations to reflect the way that you need to manage your requirements knowledge.

It is, of course, necessary for everyone using the knowledge model to have the same understanding of what the names of the classes mean. This means you need a dictionary to support the model. The following is a definition of the classes of knowledge and their associations; the classes are listed before the associations.

Knowledge Classes

Knowledge class: Atomic Requirement

Purpose

A Requirement specifies a business need; it has a number of attributes.

Attributes

Requirement Number

Requirement Description

Requirement Rationale

Requirement Type

Requirement Fit Criterion

Requirement Source

Customer Satisfaction

Customer Dissatisfaction

Conflicting Requirements

Dependent Requirements

Supporting Material

Version Number

Considerations

Also see the subtypes of requirements—namely, Constraint, Functional Requirement, Non-functional Requirement, Technological Requirement.

Suggested Implementation

Sections 9 though **17** of the Volere Requirements Specification Template. There are various automated tools available; these allow team access to the requirements.

Knowledge class: Business Event

Purpose

A Business Event is some happening outside the work scope that is, in effect, a demand for some service provided by the work. Examples: a motorist passes an electronic tollbooth, a customer orders a book, a doctor asks for the scan of a patient, a pilot lowers the landing gear.

Business events can also happen because of the passage of time. Examples: if a customer's bill is not paid in 30 days, then it is time for the work to send a reminder; it is two months before an insurance policy is due to expire.

Attributes

Business Event Name

Business Event Adjacent Systems/Actors

Business Event Summary

Considerations

It is important to recognize the business event. Its nature, the circumstances that exist at the time the event happens, and the activity of the adjacent system at the time of the business event are all important indicators of the appropriate response.

Suggested Implementation

Section 6 of the Volere Requirements Specification Template. A list of the business events and their associated input and output flows will suffice. It is practical to give each business event a unique identifier.

Knowledge class: Business Use Case

Purpose

A Business Use Case (BUC) is the processing done in response to a business event. Example: a policyholder decides to make a claim is a business event. The business use case is all the processing done by the work to approve or deny the claim. Also see Product Use Case.

Attributes

Business Use Case Name

Business Use Case Description

Business Use Case Input

Business Use Case Outputs

Business Use Case Rationale

Business Use Case Priority

Normal Case Scenario

Exception Case Scenarios

Preconditions

Post or exit conditions

Considerations

Business use cases are self-contained portions of the work, and can be studied independently. For this reason they are an important unit that project leaders can use to structure the analytical work.

Suggested Implementation

Section 6 of the Volere Requirements Specification Template. BUCs can be represented using any combination of business process models, sequence diagrams, activity diagrams, scenarios, or any other representation that is acceptable to the people involved—providing the BUC is within the boundaries declared for the Business Event.

Knowledge class: Constraint

Purpose

A Constraint is a type of requirement. It is a limit placed on the design of the product, or on the project itself, such as budget or time restrictions.

Considerations

We treat constraints as a type of requirement that must be met. However, we highlight them, as it is important that you and your management are aware of them.

Suggested Implementation

Section 3 of the Volere Requirements Specification Template. Design constraints should be recorded in the same way as the other requirements. See the Knowledge class **Requirement** for the attributes.

Knowledge class: Fact/Assumption

Purpose

An Assumption states an expectation on which decisions about the project are based. For example, it might be an assumption that another project will be finished first, or that a particular law will not be changed, or that a particular supplier will reach a specified level of performance. If an assumption turns out not to be true, there might be far-reaching and unknown effects on the project.

A Fact is some knowledge that is relevant to the project and affects its requirements and design. A Fact can also state some specific exclusion from the product and the reason for that exclusion.

Fact/Assumption is a global class and could have an Association with any of the other classes in your knowledge model.

Attributes

Description of the Assumption/Fact

Reference to people and documents for more details

Considerations

Assumptions indicate a risk. For this reason they should be highlighted and all affected parties made aware of the assumption. You could consider installing a mechanism to resolve all assumptions before implementation starts.

Suggested Implementation

Section 5 of the Volere Requirements Specification Template; these can be written in free text. They should be regularly circulated to management and the project team.

Knowledge class: Functional Requirement

Purpose

A Functional Requirement is something that the product must do. Examples: calculate the fare, analyze the chemical composition, record the change of name, find the new route. Functional requirements are concerned with creating, updating, referencing, and deleting the essential subject matter within the context of the study.

Attributes

This is a subtype of **Atomic Requirement** and inherits its attributes.

Suggested Implementation

Section 9 of the Volere Requirements Specification Template. See the Knowledge class **Atomic Requirement** for the attributes.

Knowledge class: Implementation Unit

Purpose

The unit for packaging your implementation.

Attributes

Implementation Unit Name

Considerations

This could be what your customers refer to as a “feature”; if your product is a consumer item, then it might be called a “function.” The choice of implementation unit is driven by a combination of your implementation technology and your implementation process. When you tailor this part of the knowledge model, you might find that you replace your implementation unit with several classes. The important issue is that you can unambiguously trace your implementation unit back to the relevant requirements.

Knowledge class: Naming Conventions & Data Dictionary

Purpose

A dictionary that defines the meaning of terms used within the requirements. This dictionary will be expanded throughout the project to include terms that are related to the implementation. This global class could have an association with any of the other classes in your knowledge model. Consistent use of the same terminology—as defined in the dictionary—helps to minimize misunderstandings.

Attributes

Name of the Term

Definition of the Term

Suggested Implementation

Section 4 of the Volere Requirements Specification Template; this should be in the form of a **glossary**. Along with the work context and the product context, it provides a good introduction for new team members.

Section 7 of the Volere Requirements Specification Template; there is a formal **dictionary** that defines all of the data in the inputs, outputs, and attributes within the scope of the work and the scope of the product. The dictionary provides a mechanism for connecting business terminology and implementation terminology.

Knowledge class: Non-functional Requirement

Purpose

A Non-functional Requirement is a quality that the product must have. Examples: fast, attractive, secure, customizable, maintainable, portable. Non-functional requirements types are Look and Feel, Usability, Performance and Safety, Operational Environment, Maintainability and Portability, Security, Cultural and Political, and Legal.

Attributes

This is a subtype of **Atomic Requirement** and inherits its attributes.

Considerations

The non-functional properties are important if the user or buyer is to accept the product.

Suggested Implementation

Sections 10 though **17** of the Volere Requirements Specification Template. It is vital that you give all non-functional requirements the correct fit criteria.

Knowledge class: Product Scope

Purpose

The product scope identifies the boundaries of the product that will be built. The scope is a summary of the boundaries of all the product use cases.

Attributes

User Names

User Roles

Other Adjacent Systems

Interface descriptions

Suggested Implementation

Section 8 of the Volere Requirements Specification Template. This should preferably be a diagram, either a use case diagram or a product scope model supported by a product use case summary table. Some interface descriptions might be supported by prototypes or simulations.

Knowledge class: Product Use Case

Purpose

A Product Use Case (PUC) is a functional grouping of requirements that will be implemented by the product. It is that part of the business use case that you decide to build as a product.

Attributes

Product Use Case Name

Product Use Case Identifier

Product Use Case Description

Product Use Case Users

Product Use Case Inputs

Product Use Case Outputs

Product Use Case Stories

Product Use Case Scenarios

Product Use Case Fit Criterion

Product Use Case Owner

Product Use Case Benefit

Product Use Case Priority

Suggested Implementation

Section 8 of the Volere Requirements Specification Template. The product use cases are a good mechanism for communication within the extended project team. They might take the form of models, user stories, scenarios, or anything else that suits the people involved. Whatever the type of rep-

resentation, the details of the PUC should be within the boundaries declared by the PUC inputs and outputs.

Knowledge class: Project Goal

Purpose

To understand why the company is making an investment in doing this project. There might be several project goals.

Attributes

Project Goal Description

Business Advantage

Measure of Success

Suggested Implementation

Section 1 of the Volere Requirements Specification Template. This is the basis for making decisions about scope, relevance, and priority; it is the guiding light for the project. Ideally, this goal should be defined as part of the project initiation. All project goals should be unambiguously defined and agreed to by stakeholders before putting effort into discovering detailed requirements.

Knowledge class: Stakeholder

Purpose

Identifies all the people, roles, and organizations that have an interest in the project. This population includes the project team, direct users of the product, other indirect beneficiaries of the product, specialists with technical skills needed to build the product, external organizations with rules or laws pertaining to the product, external organizations with specialist knowledge about the product's domain, opponents of the product, and producers of competitive products.

Attributes

Stakeholder Role

Stakeholder Name

Types of Knowledge

Necessary Participation

Appropriate Trawling Techniques

Contact information (e.g., e-mail address)

Suggested Implementation

Section 2 of the Volere Requirements Specification Template. Use the stakeholder map and stakeholder analysis template to define the attributes for each stakeholder.

Knowledge class: System Architecture Component

Purpose

A piece of technology, software, hardware, or abstract container that influences, facilitates, or places constraints on the design.

Knowledge class: Technological Requirement

Purpose

A Technological Requirement exists because of the technology chosen for the implementation. These requirements are there to serve the purposes of the technology, and are not originated by the business.

Attributes

This is a subtype of **Atomic Requirement** and inherits its attributes.

Considerations

The technological requirements should be considered only when you know the technological environment. They can be recorded alongside the business requirements, but it must be clear which is which.

Knowledge class: Test

Purpose

The design for Test is the result of a tester reviewing a requirement's fit criterion (precise measure) and designing a cost-effective test to prove whether a solution meets the fit criterion.

Considerations

You might consider having your testing people write the test cases as the requirements are being written. Also consider that the requirement's fit criterion is the basis of the test case.

Knowledge class: Work Scope

Purpose

Defines the boundary of the investigation necessary to discover, invent, understand, and identify the requirements for the product.

Attributes

Adjacent Systems

Input Data Flows

Output Data Flows

Work Context Description

Considerations

The work scope should be recorded publicly, as our experience indicates that it is the most widely referenced document. A context model is an effective communication tool for defining the work context.

Suggested Implementation

Section 7 of the Volere Requirements Specification Template. This is best illustrated with a context model.

Associations

Association: Business boundary

Purpose

To partition the work context according to the functional reality of the business.

Multiplicity

For each Business Event, there is one Work Context.

For each Work Context, there are potentially many Business Events.

Association: Business relevancy

Purpose

To ensure that there are relevant business connections between the scope of the investigation, the project purpose, and the stakeholders.

Multiplicity

The trinary Association is as follows:

For each instance of one Work Context and one Stakeholder, there are one or more Project Purposes.

For each instance of one Project Purpose and one Stakeholder, there is one Work Context.

For each instance of one Project Purpose and one Work Context, there are potentially many Stakeholders.

Association: Business responding

Purpose

To reveal which business use cases are used to respond to the business event.

Multiplicity

For each Business Event, there is usually one, but could be more than one, Business Use Case.

For each Business Use Case, there can be only one triggering Business Event.

Association: Business tracing

Purpose

To keep track of which requirements are generated by which business use cases. Note that this is a many-to-many association because a given requirement might exist in more than one business use case.

Multiplicity

For each Business Use Case, there are potentially many Atomic Requirements.

For each Atomic Requirement, there are potentially many Business Use Cases.

Association: Implementing

Purpose

To keep track of which product use cases are implemented in which implementation units.

Multiplicity

For each Product Use Case, there are potentially many Implementation Units.

For each Implementation Unit, there are potentially many Product Use Cases.

Association: Owning

Purpose

To keep track of which stakeholders are the originators of which requirements. The idea of "ownership" is to identify a person who takes the responsibility for helping to get answers to questions about the requirement.

Multiplicity

For each Requirement, there is one Stakeholder.

For each Stakeholder, there are potentially many Requirements.

Association: Product partitioning

Purpose

All the product use cases together form the complete scope of the product. The product scope is partitioned into a number of product use cases.

Multiplicity

For each Product Use Case, there is one Product Scope.

For each Product Scope, there are potentially many Product Use Cases.

Association: Product tracing

Purpose

To keep track of which requirements are contained in which product use cases for the purpose of traceability and dealing with change.

Multiplicity

For each Requirement, there are potentially many Product Use Cases.

For each Product Use Case, there are potentially many Atomic Requirements.

Association: Supporting

Purpose

To keep track of which systems architecture components support which implementation units for the purpose of tracking tests and assessing impact of change.

Multiplicity

For each System Architecture Component, there are potentially many Implementation Units.

For each Implementation Unit, there are potentially many System Architecture Components.

Association: Testing

Purpose

To keep track of which atomic requirements or PUC-related groups of atomic requirements are covered by which tests.

Multiplicity

For each Test, there are potentially many Atomic Requirements.

Knowledge Model Annotated with Template Section Numbers

The Volere Requirements Knowledge Model is a formal structure for identifying and relating classes of requirements knowledge. In the view of it presented in **Figure D.2**, the numbers on the classes provide cross-references to the relevant sections of the Volere Requirements Specification Template.

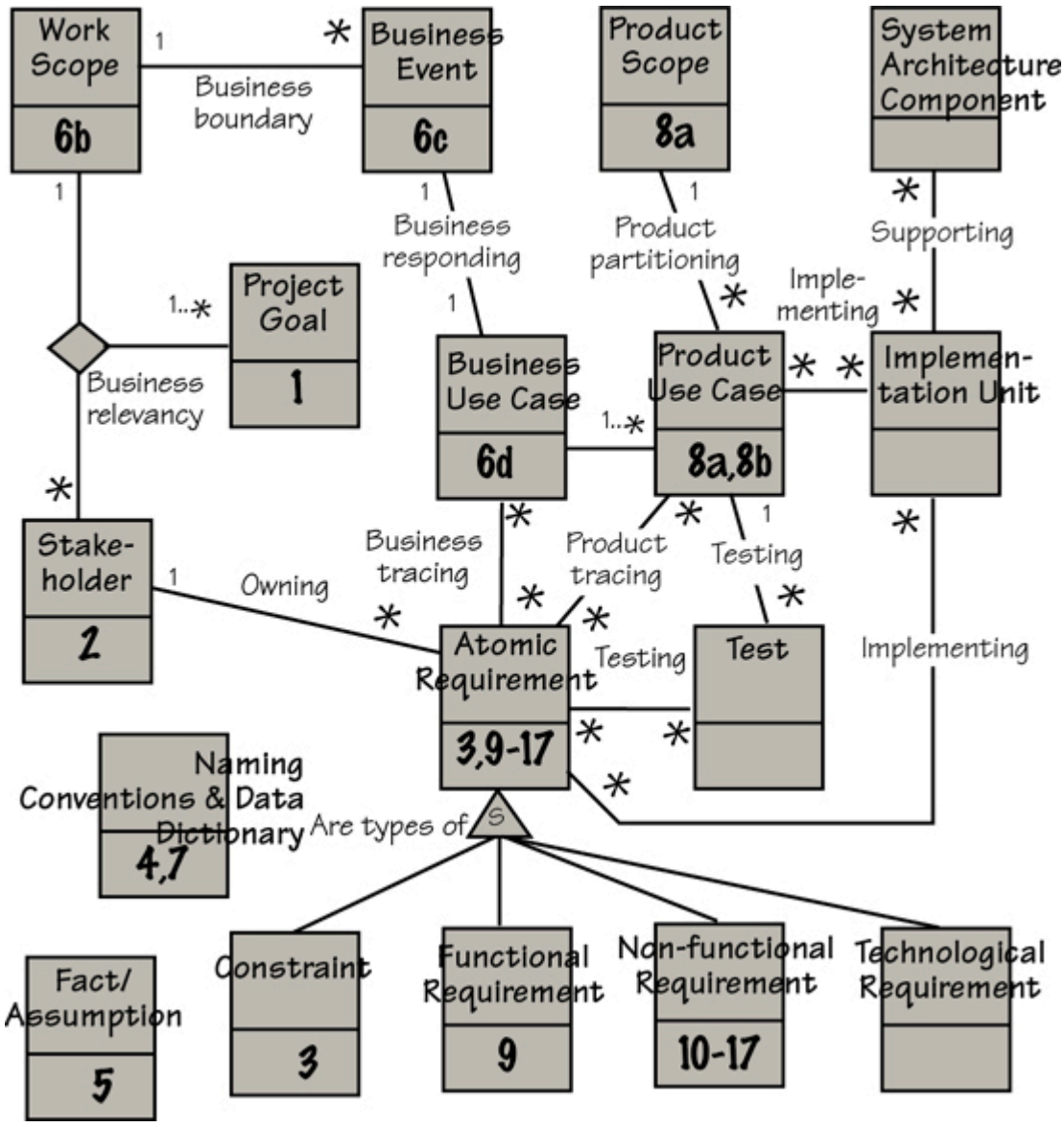

Figure D.2. The Volere Requirements Knowledge Model with cross-references to the relevant sections of the Volere Requirements Specification Template