IEC SRD 62559-4

Edition 1.0 2020-03

SYSTEMS REFERENCE DELIVERABLE

colour inside

Use case methodology –
Part 4: Best practices in use case development for IEC standardization processes and some examples for application outside standardization

IEC SRD 62559-4:2020-03(en)

About the IEC
The International Electrotechnical Commission (IEC) is the leading global organization that prepares and publishes International Standards for all electrical, electronic and related technologies.

About IEC publications
The technical content of IEC publications is kept under constant review by the IEC. Please make sure that you have the latest edition, a corrigendum or an amendment might have been published.

IEC publications search - webstore.iec.ch/advsearchform
The advanced search enables to find IEC publications by a variety of criteria (reference number, text, technical committee,…). It also gives information on projects, replaced and withdrawn publications.

IEC Just Published - webstore.iec.ch/justpublished
Stay up to date on all new IEC publications. Just Published details all new publications released. Available online and once a month by email.

IEC Customer Service Centre - webstore.iec.ch/csc
If you wish to give us your feedback on this publication or need further assistance, please contact the Customer Service Centre: sales@iec.ch.

Electropedia - www.electropedia.org
The world's leading online dictionary on electrotechnology, containing more than 22 000 terminological entries in English and French, with equivalent terms in 16 additional languages. Also known as the International Electrotechnical Vocabulary (IEV) online.

IEC Glossary - std.iec.ch/glossary
67 000 electrotechnical terminology entries in English and French extracted from the Terms and Definitions clause of IEC publications issued since 2002. Some entries have been collected from earlier publications of IEC TC 37, 77, 86 and CISPR.

IEC SRD 62559-4

Edition 1.0 2020-03

SYSTEMS REFERENCE DELIVERABLE

colour inside

Use case methodology – Part 4: Best practices in use case development for IEC standardization processes and some examples for application outside standardization

INTERNATIONAL ELECTROTECHNICAL COMMISSION

ICS 29.020

ISBN 978-2-8322-7939-7

Warning! Make sure that you obtained this publication from an authorized distributor.

CONTENTS

INTERNATIONAL ELECTROTECHNICAL COMMISSION

____

USE CASE METHODOLOGY –

Part 4: Best practices in use case development for IEC standardization processes and some examples for application outside standardization

FOREWORD

1) The International Electrotechnical Commission (IEC) is a worldwide organization for standardization comprising all national electrotechnical committees (IEC National Committees). The object of IEC is to promote international co-operation on all questions concerning standardization in the electrical and electronic fields. To this end and in addition to other activities, IEC publishes International Standards, Technical Specifications, Technical Reports, Publicly Available Specifications (PAS) and Guides (hereafter referred to as "IEC Publication(s)"). Their preparation is entrusted to technical committees; any IEC National Committee interested in the subject dealt with may participate in this preparatory work. International, governmental and non-governmental organizations liaising with the IEC also participate in this preparation. IEC collaborates closely with the International Organization for Standardization (ISO) in accordance with conditions determined by agreement between the two organizations.
2) The formal decisions or agreements of IEC on technical matters express, as nearly as possible, an international consensus of opinion on the relevant subjects since each technical committee has representation from all interested IEC National Committees.
3) IEC Publications have the form of recommendations for international use and are accepted by IEC National Committees in that sense. While all reasonable efforts are made to ensure that the technical content of IEC Publications is accurate, IEC cannot be held responsible for the way in which they are used or for any misinterpretation by any end user.
4) In order to promote international uniformity, IEC National Committees undertake to apply IEC Publications transparently to the maximum extent possible in their national and regional publications. Any divergence between any IEC Publication and the corresponding national or regional publication shall be clearly indicated in the latter.
5) IEC itself does not provide any attestation of conformity. Independent certification bodies provide conformity assessment services and, in some areas, access to IEC marks of conformity. IEC is not responsible for any services carried out by independent certification bodies.
6) All users should ensure that they have the latest edition of this publication.
7) No liability shall attach to IEC or its directors, employees, servants or agents including individual experts and members of its technical committees and IEC National Committees for any personal injury, property damage or other damage of any nature whatsoever, whether direct or indirect, or for costs (including legal fees) and expenses arising out of the publication, use of, or reliance upon, this IEC Publication or any other IEC Publications.
8) Attention is drawn to the Normative references cited in this publication. Use of the referenced publications is indispensable for the correct application of this publication.
9) Attention is drawn to the possibility that some of the elements of this IEC Publication may be the subject of patent rights. IEC shall not be held responsible for identifying any or all such patent rights.

IEC SRD 62559-4, which is a Systems Reference Deliverable, has been prepared by IEC systems committee Smart Energy.

The text of this Systems Reference Deliverable is based on the following documents:

Draft SRD	Report on voting
SyCSmartEnergy/105/DTS	SyCSmartEnergy/114/RVDTS

Full information on the voting for the approval of this Systems Reference Deliverable can be found in the report on voting indicated in the above table.

This document has been drafted in accordance with the ISO/IEC Directives, Part 2.

A list of all parts in the IEC 62559 series, published under the general title *Use case methodology*, can be found on the IEC website.

The committee has decided that the contents of this publication will remain unchanged until the stability date indicated on the IEC website under "http://webstore.iec.ch" in the data related to the specific publication. At this date, the publication will be

- reconfirmed,
- withdrawn,
- replaced by a revised edition, or
- amended.

IMPORTANT – The 'colour inside' logo on the cover page of this publication indicates that it contains colours which are considered to be useful for the correct understanding of its contents. Users should therefore print this document using a colour printer.

INTRODUCTION

0.1 General

The IEC 62559 use case template and methodology evolved from work originally performed by the Electric Power Research Institute (EPRI) as part of the IntelliGrid program. The primary purpose of that effort was to develop descriptions of existing and future power systems and their functions and requirements. In the evolution of this effort, the value of use cases as a means to accurately and completely describe the requirements for these systems and functions was demonstrated. The use case template was contributed to the IEC and this became a Publicly Available Specification (IEC PAS 62559:2008). As the best practice of use cases evolved, IEC PAS 62559:2008 was cancelled and replaced by IEC 62559-2:2015 to reflect these updates.

This methodology was originally developed as part of the IntelliGrid Architecture developed by the Electrical Power Research Institute (EPRI) as a means to implement the "IntelliGrid vision" of the automated, self-healing, and efficient power system of the future. However, the aim of IEC 62559 has changed in such a way that it is now intended to describe a methodology which is generic enough to become applicable for all domains served by IEC or other standardization bodies.

Initially, IEC 62559 was dedicated to the smart grid domain, but with the introduction of systems committees within IEC's organizational structure, the focus was widened to allow the use of the use case methodology also for other domains like active assisted living or smart cities. This document also explains how the generic methodology of IEC 62559 can be dedicated to a certain domain by complementary standards, e.g. the IEC 62913 series for smart energy [1], [2].

0.2 Objectives of this document

As defined by the IEC, the scope of IEC systems committees like Smart Energy (SyC SE), Active Assisted Living (SyC AAL) and others is to prepare and coordinate, in co-operation with IEC technical committees and subcommittees, the development of International Standards and other deliverables with emphasis on overall system aspects of technical systems and acceptable balance between cost and quality for the users of these technical systems.

While SyC SE's main focus is on standardization in the field of smart energy in order to provide systems level standardization, coordination and guidance in the areas of smart grid and smart energy, including interaction in the areas of heat and gas, SyC SE works also on methodology and tools to support the systems approach in standardization. In this regard, SyC SE has the aim to widely consult within the whole IEC community and the broader stakeholder community to provide overall systems level value, support and guidance to the TCs and other standards development groups, both inside and outside the IEC.

This document has therefore been developed to address the following objectives:

- To develop a standard methodology for determining and defining user requirements in a consistent and comprehensive manner. Standards often address only the technical issues that are included in technical specifications; however, it is just as vital to develop standards to assist users to clearly and comprehensively define their requirements.
- To clarify the distinction between "user requirements" (the "what" as needed by domain system experts) and "technical specifications" (the "how" as technical descriptions of systems, applications, and information flows to meet the "what"). Currently this distinction is an "invisible line" so that often the "what" and the "how" are mixed together – with technology-oriented project engineers jumping directly to the "how" without fully exploring the "what" with the domain system experts.

- To emphasize the critical need to determine all user requirements first, before any commitments are made on “how” to meet those requirements. Because automation and control systems are so complex and are becoming increasingly so, if all requirements are not clearly defined first, then the premature design of systems can block or seriously hinder meeting those requirements that were not initially recognized.
- To provide a means for testing the systems once implemented to ensure that the user requirements are truly met, regardless of what standards and technologies are ultimately incorporated by the vendors.

USE CASE METHODOLOGY –

Part 4: Best practices in use case development for IEC standardization processes and some examples for application outside standardization

1 Scope

This document specifies best practices for an entity to engage in a use cases redaction process to determine and describe their user requirements for systems, based on the business needs. It complements the information in IEC TR 62559-1, IEC 62559-2 [3] and IEC 62559-3 [4] by providing users with best practices in:

- use cases drafting process,
- determining the skill sets of the people required,
- use case repository management, and
- using use cases for IEC or enterprise projects.

2 Normative references

There are no normative references in this document.

3 Terms, definitions and abbreviated terms

3.1 Terms and definitions

For the purposes of this document, the following terms and definitions apply.

ISO and IEC maintain terminological databases for use in standardization at the following addresses:

- IEC Electropedia: available at http://www.electropedia.org/
- ISO Online browsing platform: available at http://www.iso.org/obp

3.1.1
use case
specification of a set of actions performed by a system, which yields an observable result that is, typically, of value for one or more actors or other stakeholders of the system

[SOURCE: ISO/IEC 19505-2:2012, 16.3.6]

3.1.2
business use case
use case that describes how business roles interact to execute a business process

Note 1 to entry: The business processes are derived from services, i.e. business transactions, which are needed to achieve different strategic goals for an organization; e.g. for the purpose of achieving specified and measurable results/products for internal or external customers.

Note 2 to entry: Business use cases are system agnostic.

[SOURCE: IEC TR 62559-1:2019, 3.8, modified – Note 2 to entry has been added.]

3.1.3
system use case
use case that describes how system and/or business roles of a given system interact to perform a function required to enable or facilitate the business processes described in business use cases

[SOURCE: IEC TR 62559-1:2019, 3.9]

3.1.4
actor
entity that communicates and interacts

Note 1 to entry: Actors can include people, software applications, systems, databases, and even the power system itself.

[SOURCE: IEC 62559-2:2015, 3.2]

3.2 Abbreviated terms

AAL	active assisted living
ATM	automated teller machine
CIM	common information model
CPU	central processing unit
RAMI 4.0	reference architectural model industry 4.0
SE	smart energy
SGAM	smart grid architecture model
SyC	systems committee
UCMR	use case management repository
UML	unified modelling language

4 Overview of the methodology

4.1 Concept of systems engineering

The use case methodology according to IEC 62559 is a subset of the science of systems engineering. Systems engineering methodology separates the concepts of “user requirements” from “technical specifications”: user requirements define “what” is needed without reference to any specific designs or technologies, while technical specifications define “how” to implement the automation systems in order to meet the user requirements.

4.2 Systems engineering methodology for use case development

4.2.1 General

The overall systems engineering methodology is illustrated in Figure 1 and consists of the following types of people and project steps.

a) Executives or other managers of an enterprise review business cases which describe and justify a perceived business need. They then approve specific projects.

b) Domain experts and project engineers are tasked to develop a project team to undertake the project. As one of the first undertakings of the project team, all domain system experts and other stakeholders (users) that could impact or be impacted by the project should be identified and represented (full time, part time, or as applicable) on the project team.

c) Domain experts review the stock of existing use cases for applicability and ideas to determine which content can inform their current effort. Preferably the use cases are stored in digital form in a so-called use case management repository (UCMR).

d) Domain experts develop a list of use cases (functional descriptions), covering not only the specific business need but other user needs and future possibilities that could impact or might be impacted by the project.

e) Domain experts, with possible assistance by project engineers who understand the use case process, draft the key use cases, capturing all the necessary user requirements.

f) Domain experts review and update these use cases to ensure their needs are captured correctly and to assess possible misunderstandings, overlaps, gaps, and other inconsistencies.

g) Project engineers assess and coordinate the use cases from which they develop a comprehensive and detailed user requirements document. This detailed user requirements document contains only user requirements.

h) Information and communication technology (ICT) specialists apply the appropriate standards and technologies, based on the user requirements document. If available, the strategic vision of a domain-specific reference architecture (e.g. IEC TR 62357-1 [5]) should be used to determine the key standards and technologies.

i) Design engineers develop the technical specifications, which combine the user requirements from the domain experts, the preferred standards and technologies from the information specialists, and the tactical approach to system development recommended by the architecture.

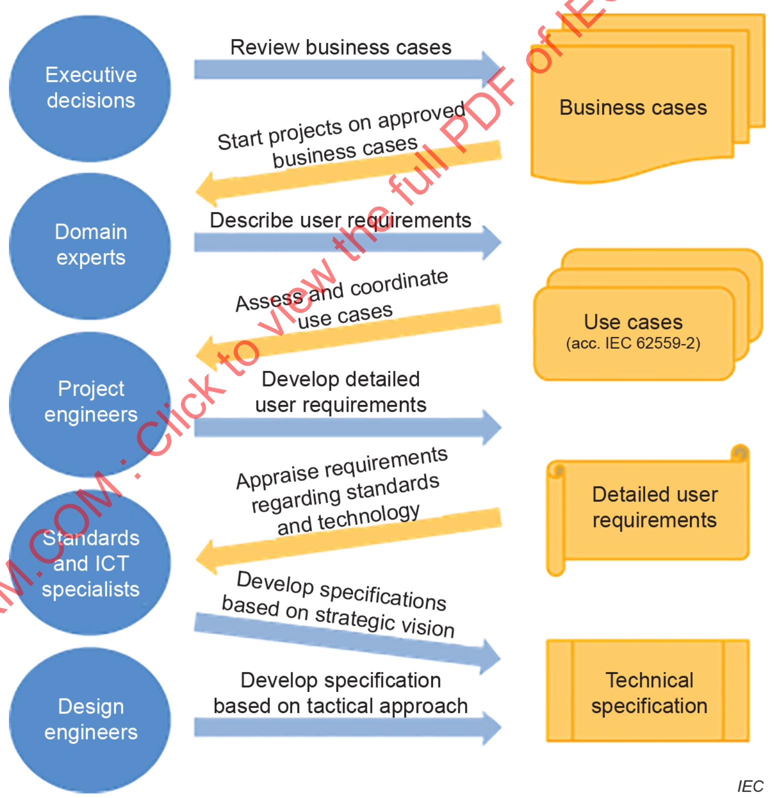


Figure 1 – Project definition process

The user requirements as elicited by the use case process and ultimately described in the detailed user requirements document cover the following aspects:

- functions from the user perspective, including functional description of processes, user choices, types of input data, types of results, and possibly display appearance;
- configuration issues, such as access to field data, electrically noisy environment, control centre LAN, or cross-organizational interactions;
- performance requirements, such as availability, response times, latency, precision, frequency of updated results, and other user parameters;

- security requirements, such as confidentiality, access restrictions, detection of failures and/or intrusions, failure management, and other safety, security, and failure issues;
- data management requirements, such as sizes, numbers of devices, amounts of data, expected growth over time, data access methods, data maintenance, and other data management considerations;
- constraints, such as contractual, legal, regulatory, safety rules, or other issues that could impact the requirements.

While a complete system engineering methodology covers both the identification of user requirements and the development of technical specifications, this document addresses only the methodology for determining and documenting the user requirements.

4.2.2 Overview of the phased approach

Although the IntelliGrid methodology, upon which this document is based, covers the entire process for developing systems, this document focuses only on the development of user requirements, and therefore concentrates on the first three phases, although it also addresses the remaining phases as they are applicable to user requirements.

Phase 1: Executives use business cases to approve projects in order to meet business needs. Although this step in the process involves executive decisions based on cost justification and other non-technical factors, from the architecture point of view, the key pre-requisite for these executives in making decisions to approve projects is that all strategic vision issues are addressed in the business cases.

Phase 2: Domain expert stakeholders give a first list of actors and describe their user requirements through the formal use case process. Use cases permit these experts to express their requirements in a formalized manner that can then be coordinated and solidified into more detailed functional and performance requirements in the next phase.

Phase 3: Project engineers develop the more detailed functional and performance requirements from the use cases that were developed by the domain experts.

Phase 4: Project engineers and IT specialists assess applicability to the project of the standards, technologies, and best practices identified in the appropriate technology review.

Phase 5: Design engineers develop the technical specifications based on strategic vision, tactical approach, applicable standards, and informed by the architecture.

4.2.3 Phase 1: Methodology for executives

4.2.3.1 Step 1: Recommendations for executives

The following are the general recommendations for executives in an enterprise.

- A domain-specific reference architecture should be adopted as a strategic vision for the enterprise information infrastructure.
- It should be ensured that the different users of the reference architecture understand how to utilize the relevant parts of reference architecture components, including functional descriptions and reference architecture strategic vision.
- A plan for implementing the reference architecture methods and standards-based technologies should be developed, based on the utility's specific business needs, the timeframe appropriate for meeting those needs, and the financial constraints.
- Feedback should be provided to the applicable standards organizations so that the standards can evolve to meet future needs and recommend standards that are created in the future.

- It should be ensured that all business cases explicitly state how the strategic vision issues will be part of the project, including use case modelling for functions, abstract data models, security issues, network and system management, data management, and integration/interoperability. Efforts that lack a clear link to strategic vision will often have trouble in being seen through to implementation and may evolve the organizational architecture in ways that are less optimal for long-term performance. If the strategic vision issues will not be part of the project, it should be ensured that the business cases explicitly state why not

4.2.3.2 Step 2: Executives and business needs

When specific business needs are identified, executives have long used business cases as the method for assessing and determining which business needs can and should be met. Business cases typically describe the business need, provide financial and organizational assessments of potential ways for meeting the business need, and recommend a specific solution with a justification for that recommendation. An organization typically lacks the resources to undertake all business cases that may be presented, and this is a necessary step in the process to ensure that only the highest value business cases are allowed to proceed.

Executives (or other decision-makers) are expected to review the business cases and approve those that meet certain justification criteria (often financial payback criteria).

4.2.3.3 Step 3: Establishing a project team

Once the executives have approved a project to meet a business need, the first step is to develop a project team. This project team should include representatives from all the main stakeholders, in order to ensure more useful functional requirements and to help ensure “buy-in” by these ultimate users of the function. Not all stakeholders need to be full-time members of the project team but should always be included in any discussions that are relevant to their areas of expertise to avoid any decisions being made in isolation or in silos, as this creates additional interfaces and jeopardizes interoperability.

4.2.3.4 Example of the outcome of Phase 1 using cash withdrawal from an ATM as business case

As most people are familiar with the use of an automated teller machine (ATM), it is a good example for explaining the steps of the several phases as a practicable example.

The business need for ATM can be summarized into two main drivers.

- Bank customers request withdrawal of money independent of the availability and opening hours of bank branches.
- Banks need to cut costs, e.g. by closing bank branches at secondary locations.

So, the challenge for a bank is to find a way to keep the customer while reducing the cost to maintain or even to improve the money-related services to the customer.

The ATM combined with service agreements within the bank sector could help to solve this challenge. However, certain standardization was paramount to enable service agreements between different bank organizations. Bank executives might issue a project to sketch out a scenario where the bank's customer may be enabled to perform a withdrawal of money at an ATM operated by a different bank (see Figure 2). This example will be used for illustration purposes through the rest of this document.

4.2.4 Phase 2: Modelling user requirements with use cases

4.2.4.1 Step 1: Identification of all potential stakeholders

One of the very first tasks of the project team should be to identify all potential stakeholders[1], even if some eventually do not directly participate in the project (see Figure 2). Often, they may have requirements that appear peripheral to the main project but could easily be met if designed in from the beginning.

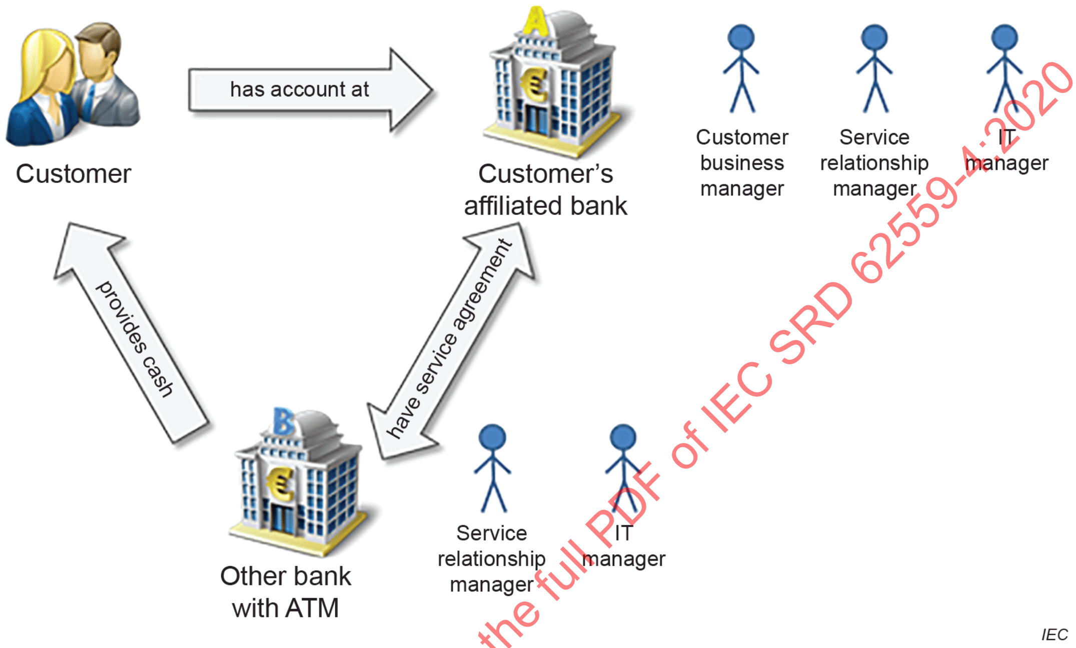


Figure 2 – Internal and external stakeholders for the ATM example

Once identified, all these stakeholders should have the project explained (briefly) to them, and then be asked if they have any user requirements that could impact (or be impacted by) the project. They should be encouraged to think "out of the box", to brainstorm future scenarios, and to envision new capabilities, rather than just restating existing functions. Thinking "out of the box" and generating "idealized designs" about what a stakeholder really needs can make profound changes in how businesses operate and how projects are implemented. This process can be difficult because understanding what might be possible under different conditions and technologies is very different from stating what is currently done.

Some of the new user requirements could just piggyback on the project without significant technical or financial impact, while others might involve changing the overall user requirements to accommodate the new needs. Other requirements might lead to simple accommodations for future expansion of the systems being implemented so that they could handle these new, but possibly not yet justified, requirements in the future. Some brainstorming discussions might cause other stakeholders to rethink their own needs in a new way.

Although not all new user requirements would be implemented immediately, this brainstorming could lead to new ways of thinking and eventually new projects to address those needs.

1 A stakeholder differs from an actor, as a stakeholder is only indirectly affected by the use case, for example the IT manager has to take care that the computer systems support all requirements of a use case.

4.2.4.2 Step 2: Reviewing repository use cases

To avoid “reinventing the wheel” the appropriate pre-existing use cases should be reviewed. There are IEC use cases managed by various working group use case managers, and there may also be parochial repositories, or a repository of use cases that already exists within the working group or technical committee.

To support the reuse, IEC is currently implementing a central use case management repository (UCMR), which will allow the collaborative editing of use cases according to the structure defined in IEC 62559-2 [3]. The UCMR will support different workspaces for different domains or systems committees, whereas each workspace could maintain its own domain-specific actor list.

In the long run, these use cases will serve as a body of institutional knowledge and serve as a checklist and a means to initiate new discussions on new functions.

4.2.4.3 Step 3: Brainstorming list of functions with stakeholders

A list of functions should be developed by the stakeholders that will capture all user requirements associated with the project area, even if some are “peripheral” to the main purpose of the project. This list of functions will ultimately be described by a set of interconnected use cases.

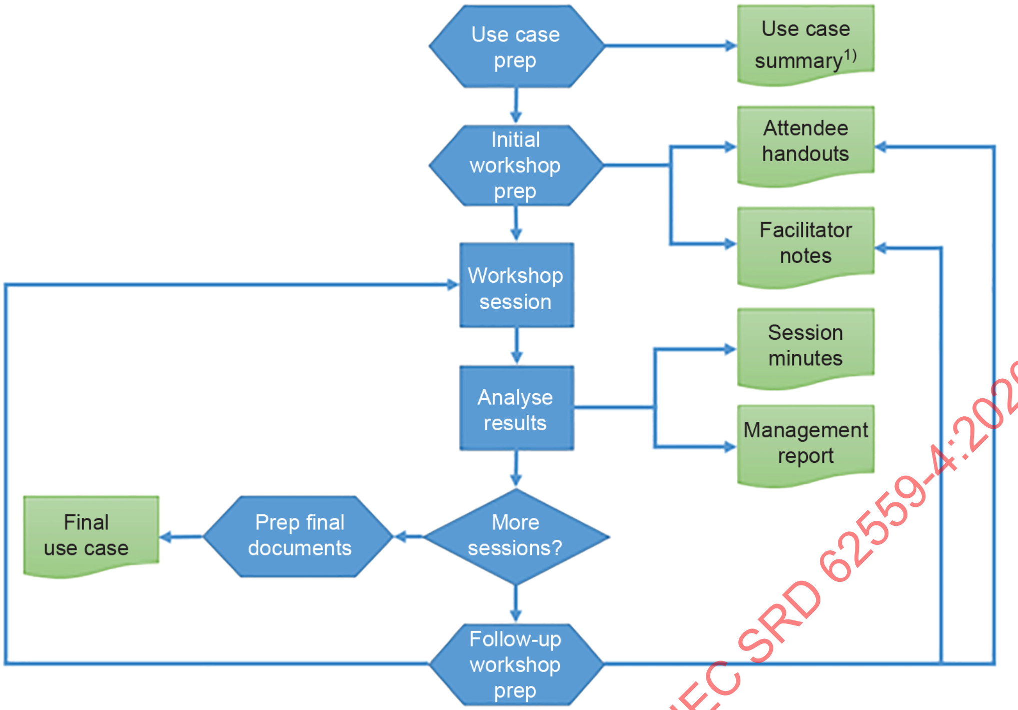
However, applying brainstorming to the use case process involves not only discarding old mindsets that inhibit creative thinking, but also learning how to use the use case process most effectively. The requirements gathering process should use an iterative and stepwise refinement-based methodology. This approach facilitates the requirements gathering process by stimulating stakeholder interest, collaborating on new ideas, and obtaining stakeholder buy-in. In some cases, the list of functions may need to be pared down, combined, or prioritized so that the primary functions are identified.

4.2.4.4 Step 4: Drafting use cases

The functions identified in the list should then be drafted into a set of use cases. These use cases should be the product of domain experts, but often these experts are not experienced in use case concepts. Therefore, project engineers who are experienced in the use case process could help elicit the requirements from the domain experts.

Drafting use cases can also be iterative, with some use cases expanded and possibly split into multiple use cases, while others are amalgamated into one (see Figure 3).

The process of translating such use case drafts into the IEC use case template is described in IEC 62559-2 [3].



[1)] Contains scenarios, actors, steps, requirements and business value statements

IEC

Figure 3 – Use case workshop process

Each requirements team executes the following procedure to define requirements while developing use cases.

1) Decide on the scope for this use case. The system will do many different things, and only certain functions will be addressed by any given use case. This will help write the narrative, which may be done now or later in the use case.
2) List the actors, what goals they want to accomplish and give a first description. Select one actor as the primary actor, whose goal will determine when the use case is done.

   Actors can be either generic or specific. For example, when modelling interactions when it is not clear what systems might be used to initiate or respond to requests, one might use a generic actor such "RequestInitiator". In other cases, the systems might be known, but specific vendors may not have been acquired so the generic system name might be used; for example, the enterprise resource planning (ERP) or the customer information system (CIS) might be used as actors. In the case where a vendor system is used, one might name the actor, for example, "Vendor A" (where Vendor A is the name of the vendor or more specifically the name of the vendor's product).
3) Identify all stakeholders and their interests. All actors representing a role are stakeholders, but there may be stakeholders who are not actors. The interests of all stakeholders need to be satisfied for the use case to be complete.
4) Identify contracts and preconditions. What has to happen before the use case can start? Knowing this will help to identify requirements.

   For each use case there will be:

   - an initial condition (the nominal state of the system);
   - a trigger (the event that initiates the need for the use case);
   - the series of steps as the use case is processed;
   - a post-condition (the state of the system once the use case has been completed).

These attributes of a use case are not only helpful in determining when a use case is complete, but can also be used to feed into a request for proposal and test criteria to determine when a vendor or the people responsible for implementing the solution have completed their work. If the post-condition (outcomes) of the use case that was documented has been met, then the work is complete. If the post-condition has not been met, then there is still work to do on the part of the vendors or solution developers.

5) Select the success scenario or the happy flow, in which the primary actor successfully achieves its goal. There may be several other scenarios in which it fails to do so.

6) Write down the steps as complete sentences written in formal language, i.e. actor-verb-object-qualifier. As each step is written, check for the following requirements and write them down:
   - What does this mean the system will have to do?
   - How quickly, reliably, safely, compatibly, useably and securely must it do it?
   - How might our business process change because of it?

7) Write the alternate scenarios. Any place that something can go wrong or differently than expected will be an alternate set of steps. Some of them may halt the main success scenario; others may re-join it later in the story, i.e. the problem could be fixed.

8) Check for completeness. Check to see if all the stakeholder's requirements, collected in phase 2; step 1, were satisfied and, in particular, if the primary actor reached its goal.

So step 4 might start with drafting a use case from the business point of view. A business use case for the ATM example is shown in a graphical presentation in Figure 4.

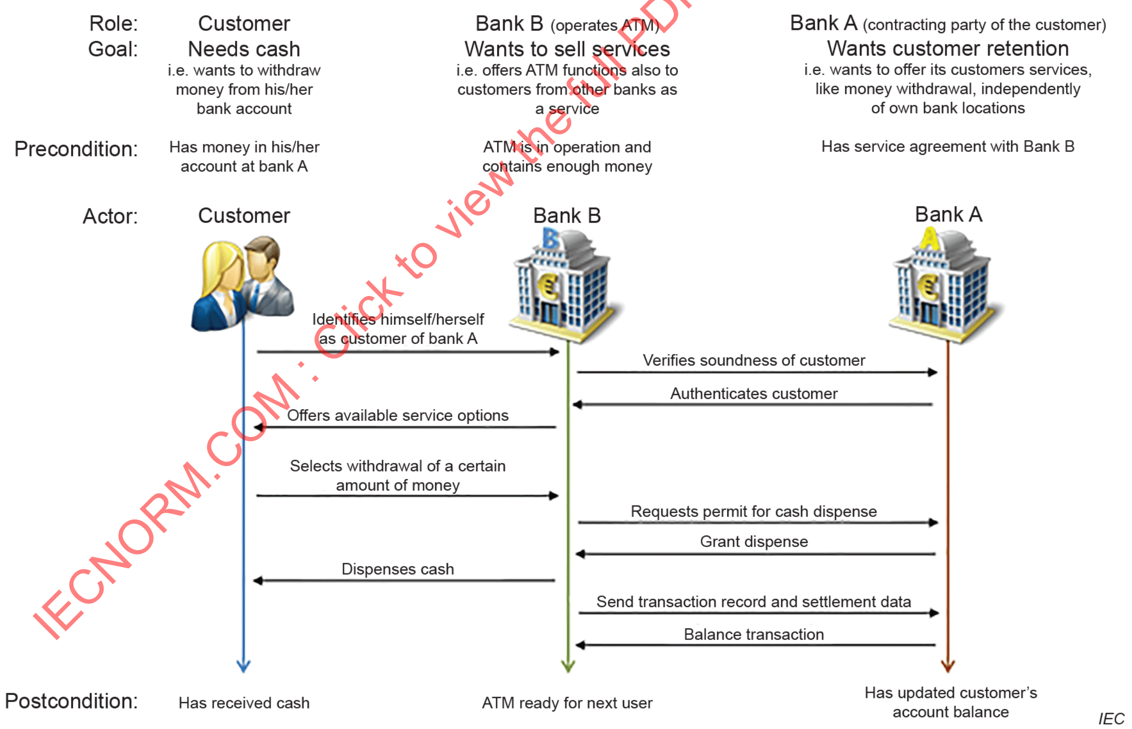


Figure 4 – Graphically drafted business use case for the ATM example

A business use case describes the interaction of business roles within the business process. For further technical analysis, the business roles might be replaced by technical systems which take over the necessary technical actions to implement the business process. This will translate a business use case into a system use case. In the ATM example, Bank A might be replaced by its bank computer and Bank B by its ATM (see Figure 5).

Also, the technical implementation of some business functions will be defined in a system use case. In the ATM example, the technical implementation of some business process functions in Figure 4 might be implemented like this.

- Customer identification → Bank card with PIN
- Presentation of available services option → Option menu on monitor screen
- Service option selection by customer → Keypad strokes

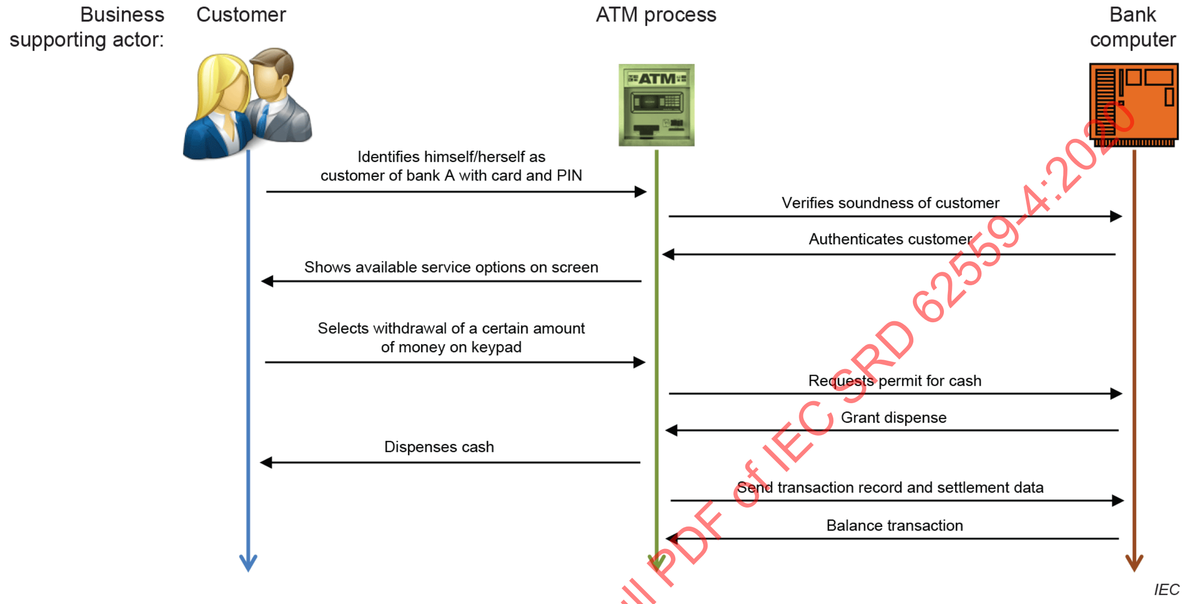


Figure 5 – Graphically drafted system use case for the ATM example

The sequence diagram in Figure 4 can easily be translated into the use case scenario steps. Each horizontal arrow specifies one step and gives the information who is sending which information to whom. Table 1 shows the IEC 62559-2 [3] compliant documentation of the ATM use case scenario steps.

Table 1 – Use case scenario description according to IEC 62559-2

Scenario								
Scenario name:	**No. 1 – Withdrawal**							
Step No.	**Event**	**Name of process/ activity**	**Description of process/activity**	**Service**	**Information producer (actor)**	**Information receiver (actor)**	**Information exchanged (IDs)**	**Requirement, R-IDs**
1	Start dialog with ATM	Identification	Customer identifies himself/herself as customer of bank	EXECUTE	Customer	ATM	Id	USE-1, AAA-1, AAA-3
2	Customer ID read	Verification	ATM verifies soundness of customer at his/her bank	GET	ATM	Bank computer	Id	AAA-3, IS-1
3	Step 2 ready	Authentication	Bank authenticates customer	REPORT	Bank computer	ATM	Auth	AAA-3, IS-1
4	Positive authentication	Service offering	ATM offers customer available service	REPORT	ATM	Customer	Menu	USE-1
5	Menu displayed	Service selection	Customer selects withdrawal of a certain amount of money	EXECUTE	Customer	ATM	Withdrawal	USE-1
6	Selection done	Permission request	The ATM requests permission from the bank to dispense cash	GET	ATM	Bank computer	Request	AAA-3, IS-1
7	Request received	Grant permission	The bank gives the permission to dispense cash	REPORT	Bank computer	ATM	Approval	AAA-3, IS,1
8	Dispense granted	Dispense cash	The ATM dispenses the requested amount of money to the customer	N.A.	ATM	Customer	N.A.	USE-1
9	Dispense done	Balancing	The ATM sends the transaction record to the bank	CREATE	ATM	Bank computer	Record	AAA-1, AAA-3, IS-1

The description of information objects exchanged in each step between two actors could be as shown in Table 2.

Table 2 – Use case information object description according to IEC 62559-2

Information exchanged			
Information exchanged, ID	**Name of information**	**Description of information exchanged**	**Requirement, R-IDs**
Id	Identification	Unique bank account identification	AAA-3, IS-1
Auth	Authentication	Confirmation of soundness of ATM user	AAA-1, IS-1
Menu	Service Menu	Display menu for selection of available options, e.g. withdrawal	USE-1
Withdrawal	Withdrawal data	Information how much money to be withdrawn from which account	USE-1
Request	Money dispense request	Request for permission to dispense requested amount of money to the customer	AAA-3, IS-1
Approval	Money dispense approval	Permission to dispense the requested amount of money to the customer	AAA-3, IS-1
Record	Transaction record	Transaction record and balancing data	AAA-2, AAA-3

Examples for the requirements which might be specified for each step or each information object could be as shown in Table 3.

Table 3 – Use case requirement description according to IEC 62559-2

Requirements (optional)		
Categories ID	**Category name for requirements**	**Category description**
AAA	Authorisation, Authentication, Accounting	Requirements for intelligently controlling access to computer resources, enforcing policies, auditing usage, and providing the information necessary to bill for services.
IS	Information Security	Information security, sometimes shortened to InfoSec, is the practice of defending information from unauthorized access, use, disclosure, disruption, modification, perusal, inspection, recording or destruction.
USE	Usability	All interaction between human user and ATM shall be intuitive.
…	…	…
Requirement ID	**Requirement name**	**Requirement description**
AAA-1	Two factor Authentication	The ATM accepts withdrawal requests only if the user enters a valid bank card and the correct corresponding personal identification number (PIN).
AAA-2	Money dispense accounting	All cash dispensing actions must be accounted correctly to allow the right balancing afterwards.
AAA-3	Unique identification	There must be a unique identification scheme for banks and bank customers
IS-1	Secure communication	All communication between the ATM and the customer's bank needs to be secured against disclosure, modification or destruction.
USE-1	User guidance	The ATM user should be guided through the withdrawal procedure in a way that the next user action is easy to understand.
…	…	…

Be sure to discuss the steps at the appropriate level, which should be at the object boundaries, not inside the object being discussed, for example, in discussing the use cases that use an ATM as an actor. What happens inside the ATM is outside the scope of the use case. The appropriate scope is the exchange of information between another system, e.g. the bank computer or the customer, and the ATM. What happens inside the ATM would only be discussed if one was designing an ATM.

Figure 6 and Figure 7 illustrate the correct way and the incorrect way to set scope, respectively.

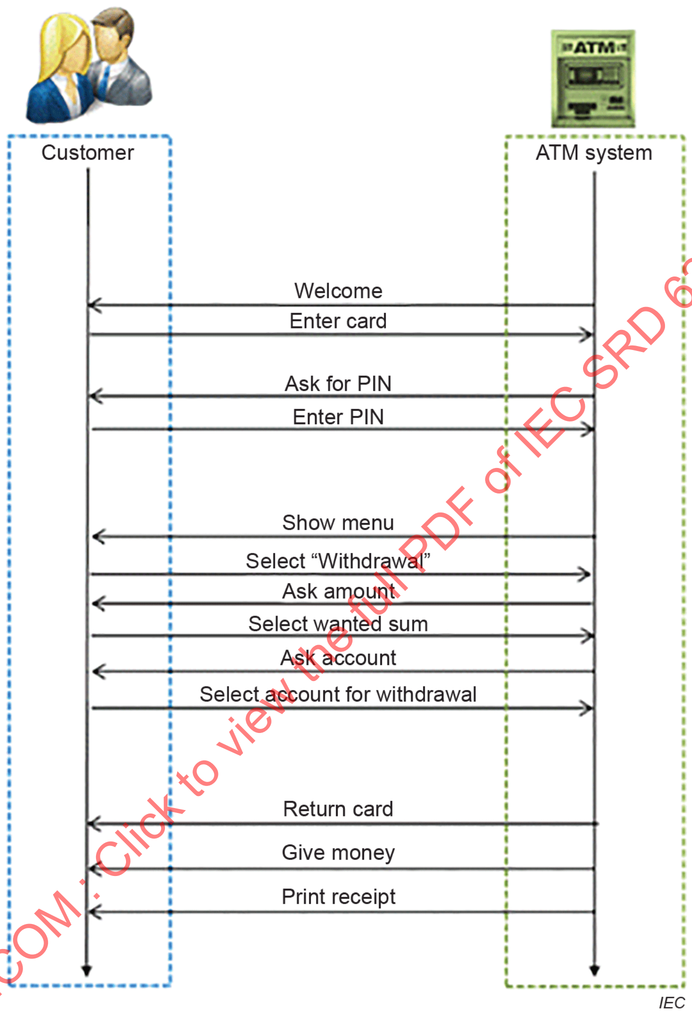


Figure 6 – Sequence diagram showing the appropriate boundaries of exchange in the step-by-step analysis

Figure 7 shows an inappropriate level of detail for a use case. Unless one was designing an ATM, the exchange of data inside any object, is not shown. The appropriate level of detail would only be at the external boundaries, i.e. the ATM is modelled as a “black box”.

In Figure 7, additional actors have been added which show internal components of an ATM: the user interface, card reader, CPU, money dispenser and printer. While potentially interesting, these are not architecturally significant as the important exchange occurs between an ATM, the customer and the bank computer, and while assessing a use case what occurs inside of the ATM is not significant.

Again, if one was designing an ATM, this might be significant. But in this example what is being designed is the exchange of data between an ATM, its user (customer) and a bank computer. In this fashion it is important to consider the scope of a use case and model the step-by-step analysis at the appropriate level.

Keeping domain experts at the appropriate level and on track is the responsibility of the use case session facilitator.

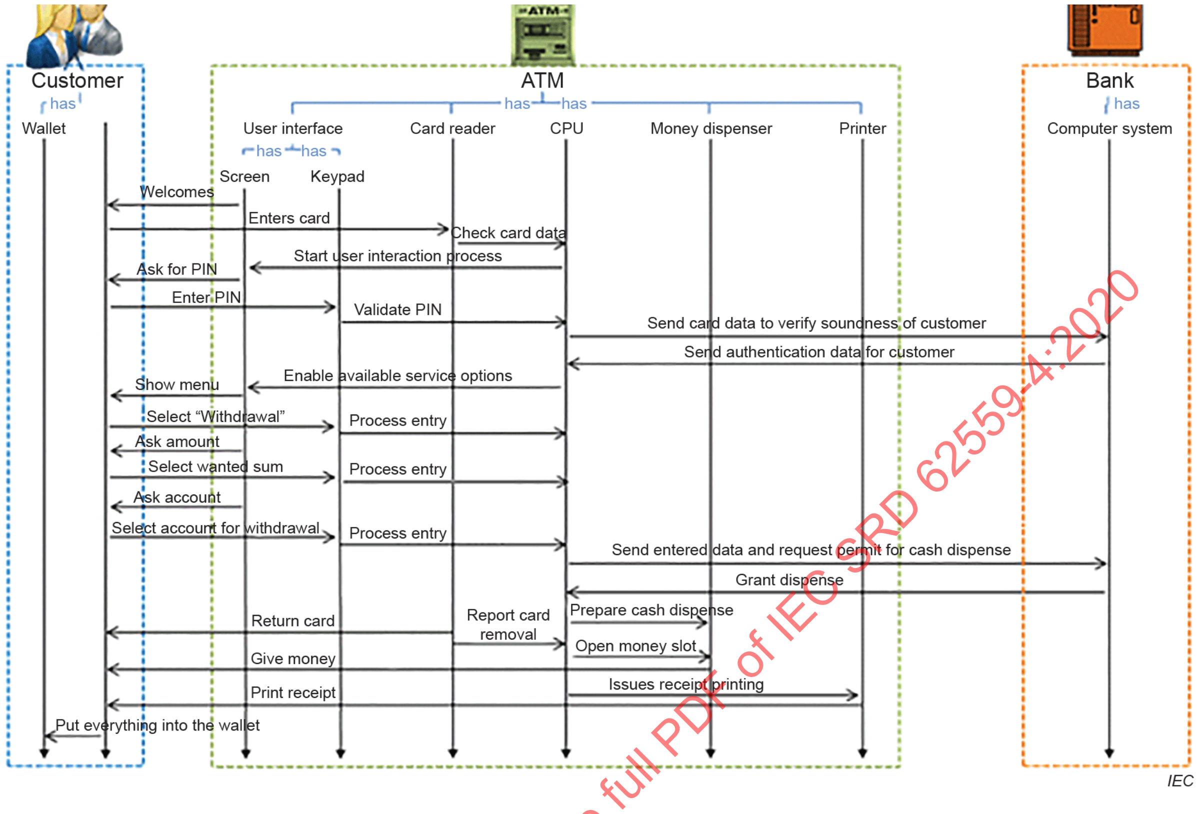


Figure 7 – Sequence diagram showing an inappropriate level of detail of information exchange

Technical processes do not always run though without error condition. Therefore, alternate scenarios might be specified, which describe the behaviour of the system if error conditions occur. Figure 8 shows possible error scenarios for the ATM.

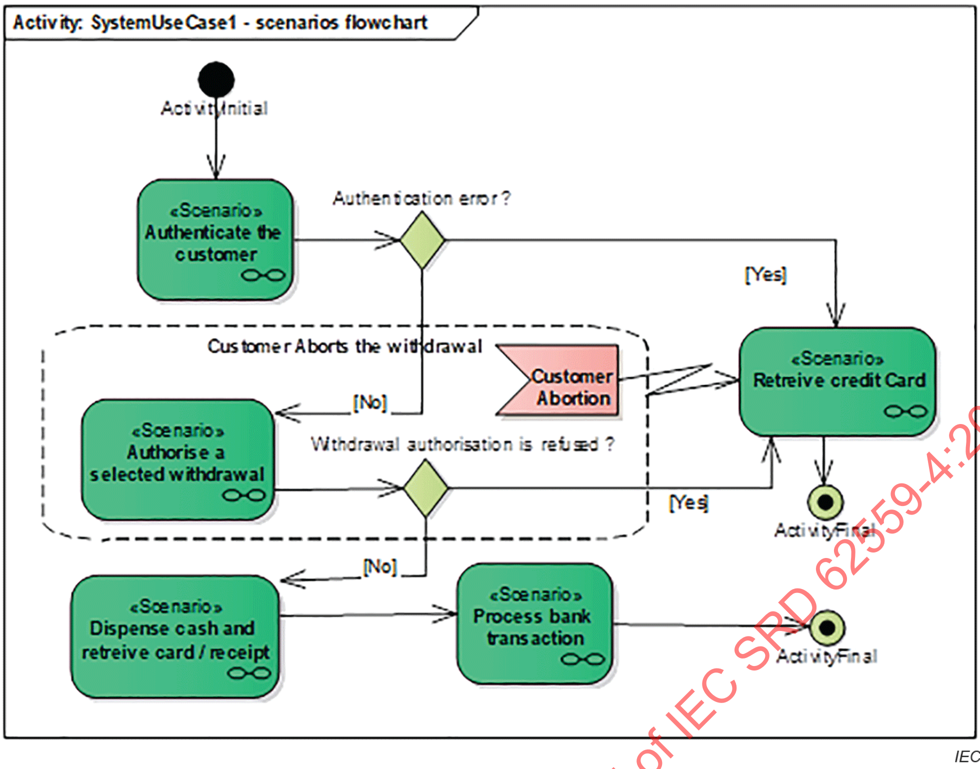


Figure 8 – Flow diagram for alternate (error) scenarios

4.2.4.5 Step 5: Reviewing and updating use cases

All domain experts and stakeholders should have a chance to review and comment on use cases. Some stakeholders may be more proactive than others, but care should be taken that the requirements of less active stakeholders are not lost. To ensure that this happens, use case session should be led by someone skilled in facilitating requirements sessions.

Some use cases may end up being split into multiple use cases, while other use cases may be combined during this process. Use cases can be updated multiple times if needed – but the decision when to stop modifying a use case can be more of an art than a science.

4.2.5 Phase 3: Methodology for project engineers: developing detailed user requirements

4.2.5.1 Step 1: Coordinating and combining use cases

Project engineers should coordinate the many use cases from the domain experts and possibly combine any common components into "subroutine" use cases, while still maintaining the unique components. The results should be reviewed by the domain experts to ensure their requirements did not get left out by accident.

NOTE The term subroutine is taken from the software development field, to illustrate the idea of referring from a high-level use case to some low-level use case to specify more details off a single high-level use case step. However, the use case methodology does not support software programming related features like parameter lists, etc.

In particular, project engineers should review the characteristics of common components (e.g. types of data, configuration, quality of service, security, and data management), and develop comprehensive and/or coordinated requirements across all use cases. For instance, they should identify the most "constraining" requirements, such as the highest level of security needed or the most rapid response requirements, so that either all elements will meet that constraint or the constrained elements are isolated from the other elements.

The list of actors recorded in the use cases typically contains many duplicates and many different names for the same actor. The systems architecture team examines the list, eliminates duplicates, agrees on common names, eliminates some actors, and establishes the scope of the architecture being developed. The resulting global actor list will be used as a common language for the rest of the architecture development.

4.2.5.2 Step 2: Developing user requirements from the use cases

Use cases are vital to understanding the individual user requirements but are difficult to view in combination. Therefore, once the individual use cases have been finalized by the domain experts and the project engineers, a single (or just a few) functional requirements documents should be developed that capture all of the (coordinated) user requirements. Just like the use cases themselves, these functional requirements address "what" is needed, but not "how" it is to be provided.

While most of the attention is given to the functional requirements, there are two types of requirements.

1) Functional requirements identify what the system must do: features, options, controls, configuration, and behaviour. They generally are either "yes" or "no" statements: Does it do the job or not?
2) Non-functional requirements identify what the system must be. They are also called constraints, behaviour, criteria, performance targets, etc. They set limits or controls on how well the system performs the functional requirements. These are the "…itys" and "…ances": reliability, security, usability, upgradeability, expandability, scalability, compatibility, safety, performance, conformance, etc. They generally have a numerical target threshold.

Functional requirements form the basis for the technical specifications which can add additional specific technical requirements.

A good set of requirements not only defines what to do, but also when the job will be finished.

It should be remembered that the development of use cases is a means to an end. The "end" is to get to a set of well-formed requirements.

Some challenges with having poorly formed requirements are:

- resources squandered on unimportant features;
- incomprehensible user interfaces;
- incompatible interfaces and products;
- missing architectural elements;
- unsafe user environments;
- un-testable features;
- rework when the error is discovered.

Table 4 – Examples of poorly-formed requirements and requisite improvement

Bad requirement	Improved requirement	Reason for improvement
"The database shall process all common file formats."	"The database shall permit administration team members to import records using any of the following file formats…"	Specifies: • who is given the capability; • which particular file formats.
"The user interface will be easy to use."	"The operator shall control the device using a Windows XP graphical user interface complying with usability standard XXYYZ."	Provides an objective measure for the vague terminology "easy to use".
"Messages shall be exchanged immediately when an alarm condition occurs."	"The server shall notify all connected client devices of an alarm condition within five seconds."	Specifies: • which system is responsible for sending the messages; • which systems shall receive the messages; • what the target time interval shall be.
"The device must contain batteries."	"The device shall retain all data and time information for at least 24 hours after a power failure."	Specifies "what", not "how". Does not specify a particular implementation (batteries), but the real requirement – namely, that data not be lost.

There are "good" requirements and "bad" requirements, some examples are given in Table 4. A good requirement clearly identifies which part of the organization is responsible for the requirement, and what it must do. It should identify "what", not "how".

ISO/IEC/IEEE 29148 [6] defines a "well-formed" requirement as a statement that:

a) states system functionality (a capability);
b) can be validated;
c) must be met or possessed by a system;
d) solves a customer problem;
e) achieves a customer objective;
f) is qualified by measurable conditions and bounded by constraints.

Specifically, a well-formed requirement should contain:

– capability;
– condition(s);
– constraint(s).

4.2.5.3 Step 3: Numbering and identification

As any given repository of use cases grows to describe a domain, for example an electrical distribution system, the number of use cases will grow into the hundreds. A mechanism will be needed to both organize and number the use cases so that it will be easier to locate a given use case in a repository. Use cases can be categorized by actor or system and should be numbered.

One way a numbering scheme could be derived is to consider the domain and the instance of a use case. For example, the Smart Grid Architecture Model (SGAM), identifies several domains and zones seen in Figure 9. One naming convention might use DOM-ZON-xxx where DOM is the domain (GEN, TRN, DST, DER, or CST for generation, transmission, distribution, distributed energy resources, and customer, respectively), ZON is the zone (PRO, FLD, STT, OPN, ENT, MKT for process, field, station, operation, enterprise, and market, respectively), and XXX is simply the instance of the use case; for example, the first use case developed would be 001.

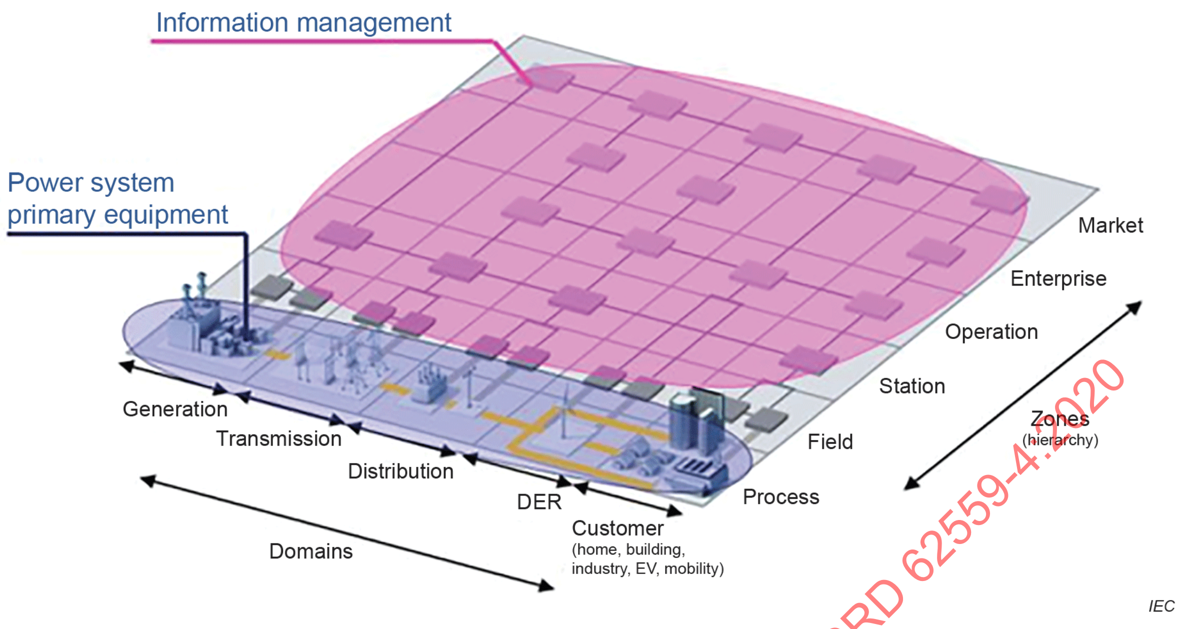


Figure 9 – Base plane of the SGAM showing its requisite domains and zones

So, a use case distribution domain that is concerned with enterprise integration and which is the twelfth such use case, might have a use case identifier of DST-ENT-012. The identifier in this case not only uniquely identifies the use case, making it easier to find in a large repository, but for people familiar with this architecture model, foreshadows the content of use case by identifying the domain and zone of the scope.

As requirements are identified they should be numbered. As they are traced through development it becomes easier to locate use cases and functional requirements in any repository, and to trace when, how, and where they have been addressed in the implementation phase and may be used in the contract language for a request for proposal.

As each requirement is identified, record it in formal requirements language, i.e. subject-verb-object-qualifier.

Also, as each requirement is identified, qualitatively identify the business value of each requirement. For instance, “Will reduce cost by eliminating the following manual steps…”. These qualitative business value statements will be used later in the business case analysis. The scenarios, actors, steps, requirements, and business value statements are the outputs from the use case process, as shown at the top of Figure 3.

4.3 Use of Unified Modelling Language (UML)

4.3.1 General

Unified Modelling Language (UML) is a way of visually representing various artefacts. While not required in the use of the use case template, they can help with the readers understanding of a use case. There are primary representations that are used in relation to use cases that are developed by the IEC:

- use case diagram (4.3.2);
- scenario flow chart (4.3.3);
- activity diagram (4.3.4);
- sequence diagram (4.3.5);
- data class diagram (4.3.6).

4.3.2 Use case diagram

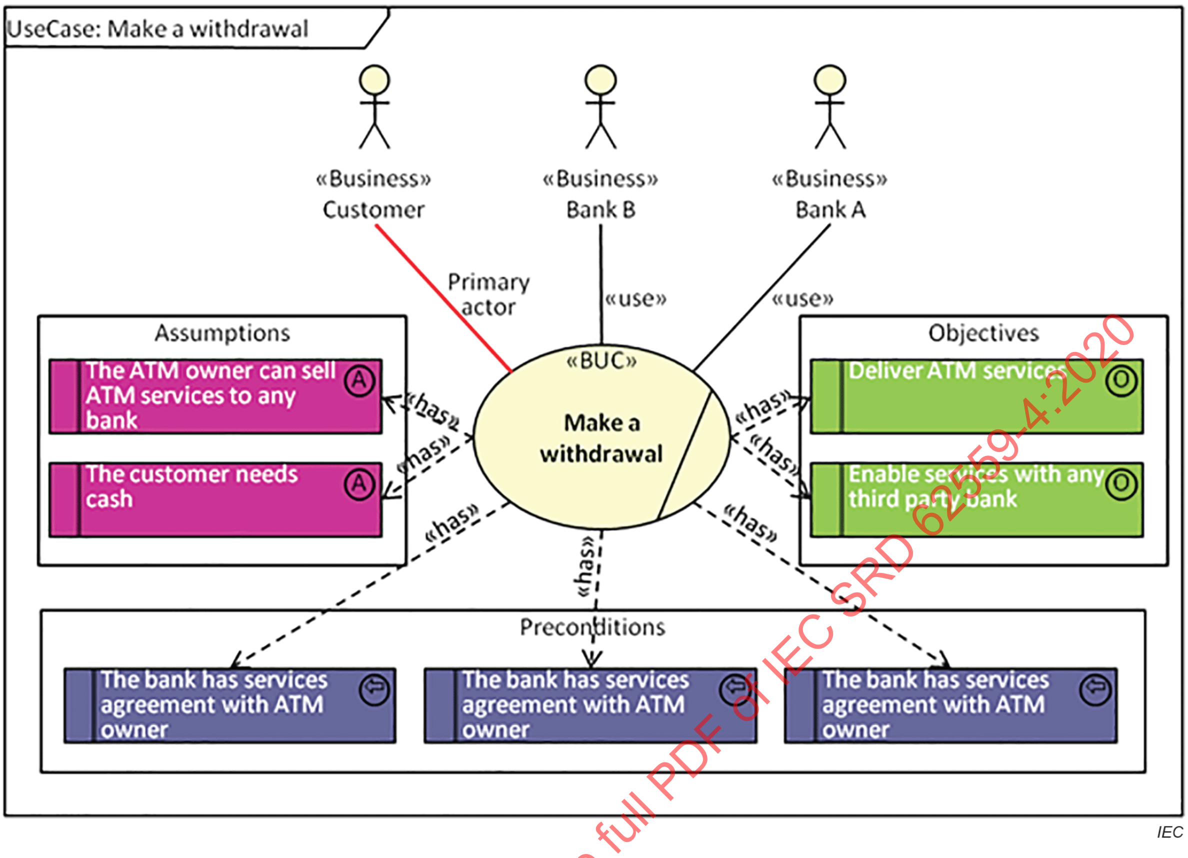


Figure 10 – Example use case diagram showing actors, assumptions, objectives and preconditions.

The use case diagram (Figure 10) can show the use case under consideration, the primary actor and all other actors involved in the use case, and optionally, assumptions, objectives and preconditions related to the use cases.

4.3.3 Scenario flow chart

A scenario flow chart (Figure 11) will show the relationship between the scenarios within one use case.

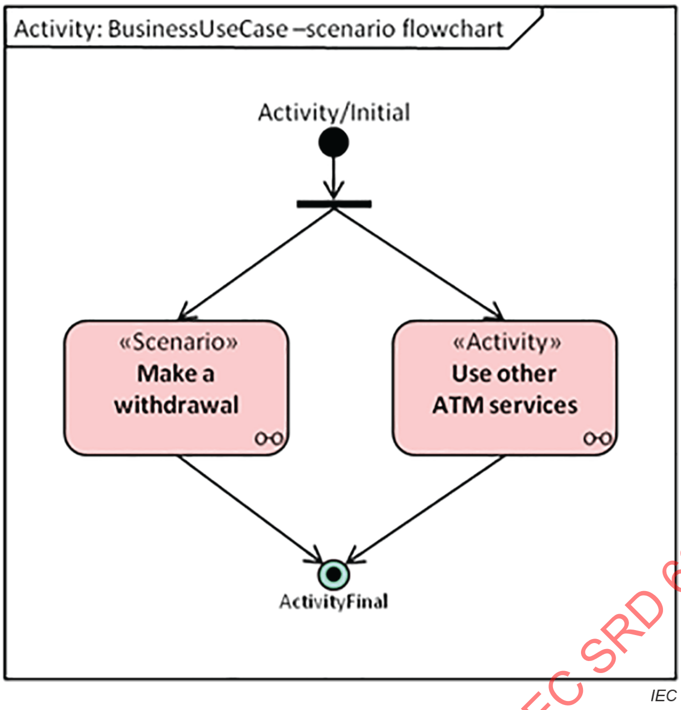


Figure 11 – Example use case diagram scenario flow

4.3.4 Activity diagram

The activity diagram shows the activity that occurs within a use case. This matches to the step-by-step analysis of a use case portion of the use case template. This shows each of the actors in a separate "swim lane" as shown in Figure 12.

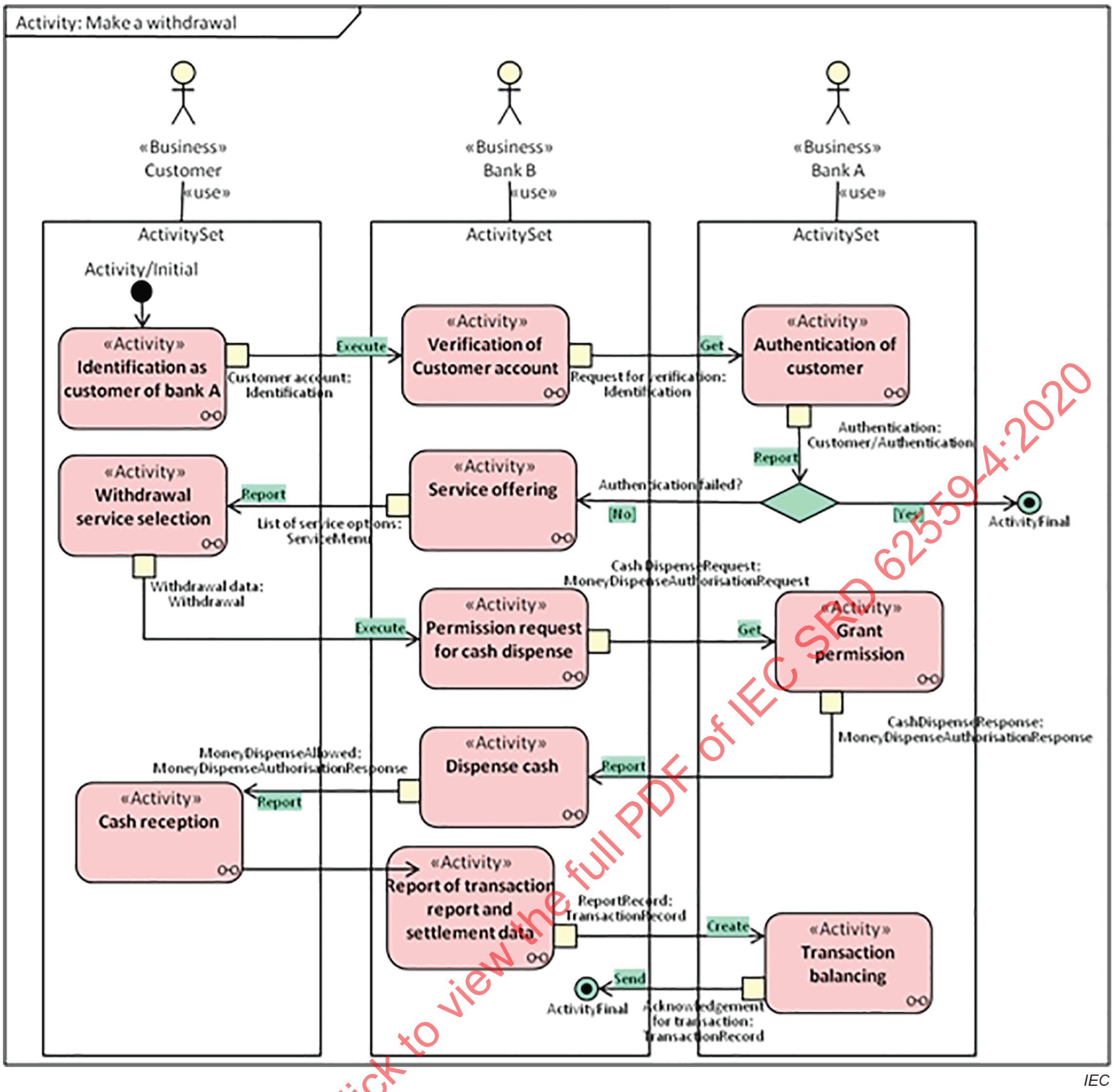


Figure 12 – Example activity diagram using Unified Modelling Language showing the interaction of ATM operator bank, contractual bank, and a customer

A useful modification to the activity diagram is the use of object flow instead of just control flow. Control flow (the arrows) only shows the transfer of control from one activity to the next. Object flow, in addition to showing the change of control, can also inform the reader of what information is being passed. This is useful because this object can then be used to link to a data class diagram and gives the traceability from use case, to requirements, to data. For example, for use cases that leverage a standardized data model like the Common Information Model (CIM), the data object can then be used to tie to a specific profile within the CIM.

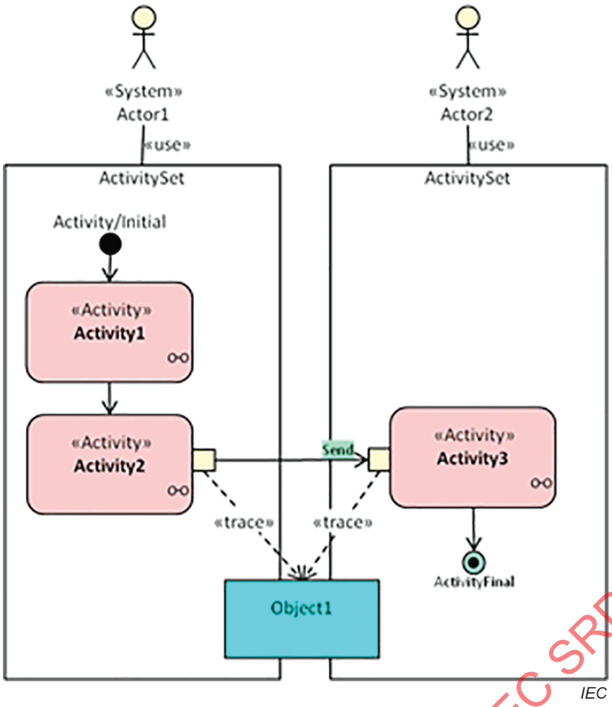


Figure 13 – Example activity diagram with object flow

The example in Figure 13 shows control flow between Activity1 and Activity2. Between Activity2 and Activity3 object flow is used. When a hand-off occurs between two actors (shown in the swim lanes), typically this involves the exchange of information. This exchange should be captured in the use case template but can also be represented in UML. With object flow, nodes will be added. In UML, if the service is known this can be captured by changing the name of the object node to the service name. For example, if using IEC 61968-1 [7], the naming would use the verb-noun naming convention, for example, MoneyDispenseAuthorisationRequest. Additionally, by using the object element, this could also be used to link to a specific profile, for example, in the CIM the object could then be linked to the appropriate data class diagram as shown in Figure 14.

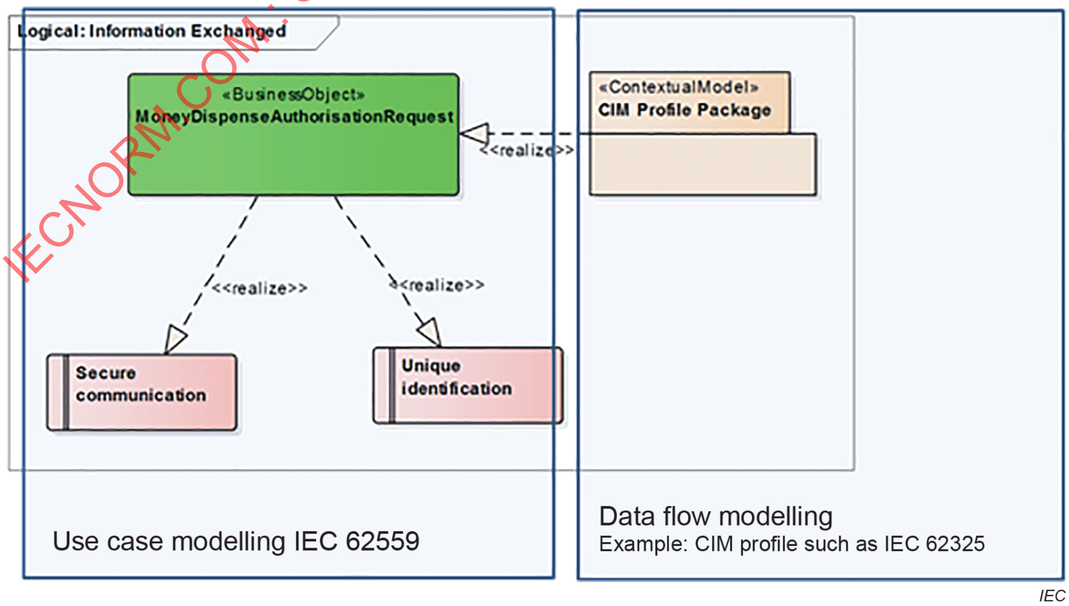


Figure 14 – Example for use case modelling down to data flow modelling

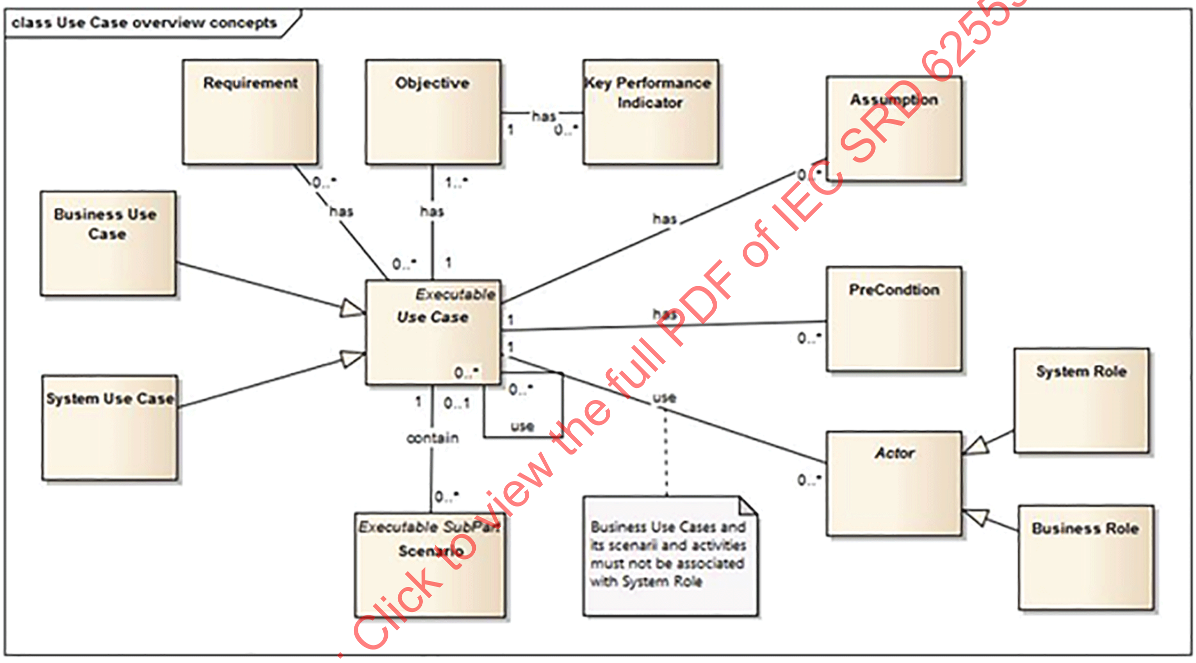
Similar approaches can also be found in other domains, for example the Banking Industry Architecture Network (BIAN) [8] or the Building Information Model (BIM) [9].

4.3.5 Sequence diagram

A sequence diagram is a type of UML diagram that decomposes the relationships between actors into only the data exchanges, showing the sequence of exchange from top to bottom. Examples for sequence diagrams are shown in Figure 6 and Figure 7.

4.3.6 Data class diagram

A data class diagram shows the classes, attributes, associations, and inheritance of related information. In the use case template this is one way to capture the data requirements that are the result of a use case. An example is shown in Figure 15. This data class diagram captures the data classes which could be derived out of the IEC 62559-2 use case template structure.



IEC

Figure 15 – Example UML data class diagram showing the elements of a use case

These UML diagrams are a helpful means to represent portions of a use case that may be of interest to particular stakeholders. The use case diagram may be of concern only to business executives, the activity diagram to the stakeholders that perform the work, sequence diagrams for the stakeholders that have to design the system integration, and the data class diagram for system integrators and data architects.

4.4 Determining the quality of a use case

4.4.1 General

Quality can be in the eye of the beholder. It is important to remember that using use cases for systems development is a means to an end, not the end. The reason that use cases are developed is so that system requirements are defined to a sufficient level of detail that mistakes or issues are discovered in the planning and design phase, not in the implementation phase. However, to that end there are questions that can be asked to determine the quality level of the use case. The following are some examples.

- Does the use case have the four parts (initial condition, trigger, steps, post-condition)?

- Are the use case actors from a standard list? If a new actor has been introduced, has it been reconciled with the standard list?
- Have the information requirements been defined to a sufficient level of detail that a profile can be designed?
- Are the requirements well-formed?
- How many empty fields exist in the use case template?
- Does each requirement have a unique ID so that it can be traced throughout the lifecycle of the project?

Additional domain-specific quality criteria may be derived from corresponding standards, for example, IEC 62913 for smart energy.

The answers to these questions will help the use case manager determine if additional work needs to be accomplished before the use case can be determined to be complete.

4.4.2 Considerations for use case management and collaborative working

As described in the beginning of the document, use case writing is a collaborative task of various experts covering all aspects to be considered for a successful technical implementation of a promising business case. While starting with a small team of authors, a use case might be distributed to larger groups of experts over its lifecycle. During the use case creation, the authors might ask additional experts to provide their feedback. If the use case authors consider their work ready for publishing, they might ask other experts for editorial quality checks or for the approval of the use case. Figure 16 shows the evolution of a use case, which should be supported by a use case management repository (UCMR).

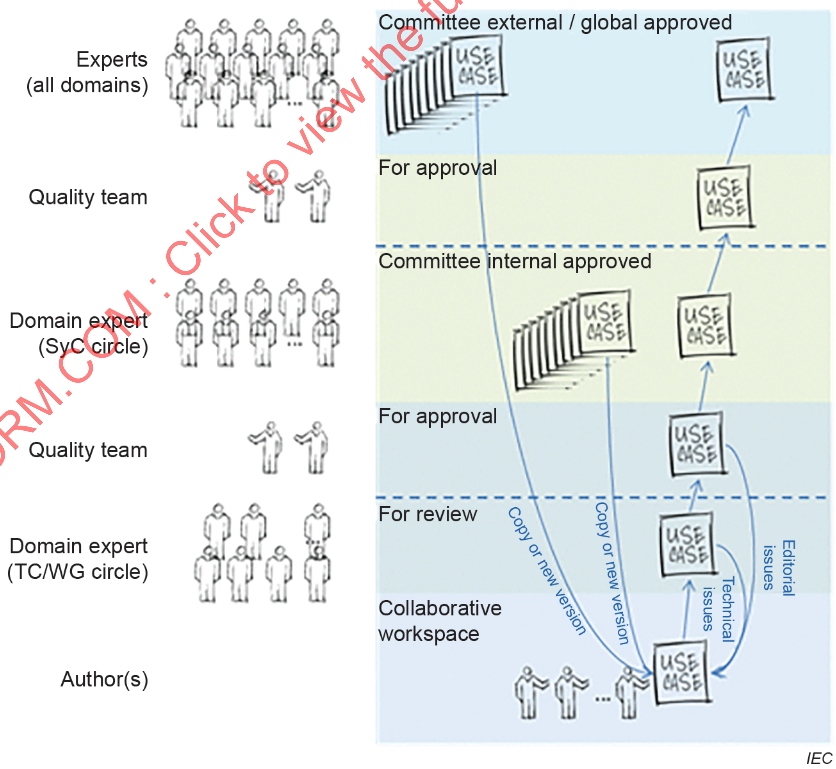


Figure 16 – Use case lifecycle process

Use case writing might start from scratch; however, the authors might also use an existing use case as starting point and create a new version or a new variant. So, versioning of a use case should also be supported by a UCMR.