NS Passenger

Rolling Stock & Energy
Projects & Project Support

Requirement Writing with Logic

Manual for the Construction
of Rolling Stock Requirement Specifications

Contents

1 Terminology

In this chapter we will explain a number of terms used in the text, using examples and presenting strategies how to recognise clauses. This chapter can be used for reference purposes. To start with a definition which is already important in this section: the symbol := means “is defined as”.

- The Arity of a connective, or relation := the number of objects bound by a particular connective or relation. A binary connective binds two sentences, and n-ary connective binds n-objects. Similarly, for the relations.
- Antecedent := the first clause of a conditional sentence. In the case “If A then B”, we say that A is the antecedent of the consequent B, as such we know that A precedes, or is the ancestor of B.
- Binary connectives := words or phrases to make one (complex) sentence out of two sentences. Examples are and, but, however, if, or, unless, before, because, since, etc.
- Consequent := the second clause of a conditional sentence. In the case “If A then B”, we say that B is the consequent of the antecedent A.
- ‘Connective’ or ‘main operation’ := operations used to link simple sentences to form complex sentences.
- Conjunction := a word for the sentences formed using the “_and_” connective.
- Conjunct := one of the elements of a Conjunction.
- Conditional := a word for the sentence formed using the “if_then_” connective.
- Disjunction := a word for the sentence formed using the “_or_” connective.
- Disjunct := one of the elements of a Disjunction.
- Formal translation := the procedure of identifying the structure of a natural language sentence and representing it in our language.
- Implication := another word for the “If_then” connective.
- Negation := a word for the “not_” connective.
- Property := a characteristic that may belong to one or many things at the same time. A property has an extension, that is, a class of things to which it belongs. Different properties may characterize the same extensions (examples are the properties “having a heart” and “having a liver”). A property may also be called a feature, quality or attribute of some individual or group of individuals
- Predicate := the name of a property or relation.
- Predicate-Object Structure := a word for the only admissible atomic sentence structure. We also call this a Function-argument structure.
- Quantifier := a word for describing any quantificational operation. We primarily discuss the existential quantifier and the universal quantifier. These notions are defined in detail below.
- Relation := a property of ordered constituents. A binary relation is a property of ordered pairs. A n-ary relation is a property of a ordered group of n-elements or individuals.
- Syntax := the study of grammatical rules of composition.
- Semantics := the study of meaning.
- Substitution := the operation of uniformly removing and replacing some signs occurring in a sentence with others in such a way that the overall truth-value of a sentence does not

change. For example; “Bugs Bunny doesn’t appear in the Bible” can be transformed by substitution into “Santa Clause doesn’t appear in the Bible” because we have uniformly swapped each occurrence of “Bugs Bunny” for “Santa Clause” and the sentence remains true. We can substitute names, predicates or variables.

- Truth-value := every sentence is defined as having a truth value (i.e. either true or false) on the basis of the facts and the semantics of the sentence.
- Unary connectives := Unary connectives are connectives which bind one sentence. Unary connectives can also be thought of as sentence modifiers, examples include not, maybe, possibly, necessarily, may, can, could, must, should, etc.
- Universe of discourse := the set of objects we could possible speak of.
- Variable := a sign with an unspecified value.

2 Introduction

The basic goal of requirements and requirement specifications is to align the expectations of NS as a customer with production design and quality as offered by suppliers of rolling stock. In order to do so, NS has to be able to describe (or prescribe if need be) its needs and desires in such a way, that client and supplier share an identical (or at least similar) image of the final product. That's where this manual comes in: it provides rules for the construction of requirements and requirement specifications for rolling stock of NS Passengers that leave little or no room for unexpected interpretations of suppliers. As such it can be considered as the requirement specification for requirement specifications.

This manual aims to create a better match between the stakeholder's needs and the supplier's solutions. To do so

1. A semi-formal language including grammar and semantics will be provided, as well as specific rules about how to evaluate any sentence of that language.
   In turns out that virtually every sentence in natural language can be translated into our semi-formal language, and thus there are simple rules to determine whether or not any natural language sentence is true.
2. A standard for structuring requirement specifications will be provided.
   The idea is that integrating the requirements properly in the specification body will serve the representation of the expectations of NS.

> In short, by writing our requirement specifications in such a way so that anyone may uniformly determine the conditions under which each requirement has been fulfilled or failed, we can expect a greater degree of convergence between the requirements of NS and the actions of the supplier. This is a benefit to both NS and any potential supplier.

We will start in chapter (3) with a broad motivation for pursuing our proposed methods, including examples from the drafts of SNG and the EuroSpec toilet specification. Chapter (4) will introduce our semi-formal language. The following chapter will elaborate the grammar of our language, before we turn, in chapter (6), to the topic of the language's semantics. In chapter (7) we combine the principles elaborated in chapters (3), (4) and (5) providing the reader with a step by step procedure for evaluating any requirement for compliance. In chapter (8) we highlight the core principle of this method. In chapter (9) we zoom out to consider the method of structuring a requirement specification logically. You can consult a glossary of terminology at the beginning of the document, in chapter (2), whenever the need arises.

3 Motivation: structuring sentences

In this chapter we will show that there are a number of requirements found in either the SNG spec, or the EuroSpec document that can be improved. Our aim is to provide you with the skills for determining why and how a sentence can be improved whether in English or in Dutch. We will show how systematic sentence construction can help to improve the quality and lucidity of requirements. Maybe even more important the logical build up of requirements make it possible to prove or disprove compliancy of the deliverables with the requirement. However, these latter considerations will be discussed in chapters (5), (6) and (7). For now we just argue for some amendments, but we don't delve too deeply into the reasoning.

The idea is to present intuitively agreeable rewrites of the requirements, and then develop a system (in chapters (4), (5), (6) and (7)) for writing requirements, and show how our proposed rewrites fall out naturally as a simple consequence of the system. This way, even if your intuitions are such that you disagree with the proposed rewrite, the fact that they are the consequence of a beautiful, concise and (most importantly) correct system of reasoning should convince you that the proposed rewrites are indeed the correct rewrites.

Examples

We now consider examples of requirements and provide minimal suggestions about how to rewrite them. We write the original requirements as option (a), followed by suggestions for improvement (b). The idea is to return to these examples towards the end of the manual, in chapter (8) and examine them with an eye to their logical structure.

1a The passenger emergency brake system shall, in the case of a passenger's emergency brake application, indicate in the cab which emergency brake handle has been operated.[1]

1b If and when the passenger emergency brake has been triggered, it shall be shown in the cab, that (a) an emergency brake has been triggered and (b) which emergency brake was triggered.

There are some unnatural features of (1a). Most importantly, the conditional sentence starts with an object and proceeds to specify conditions about it, but it should state the conditions of the object and the generally required features of the design, when such an object is in those conditions. The latter option has the benefit of conveying the wholly general nature of our stipulation. The original format relies on the definition of "shall" to convey such information, but while this is not wrong it is less forceful than the alternative, which maximizes the strength of the imperative by coupling the use of "shall" with the appropriate sentence structure.

2a. The toilet system shall be able to operate within 1 hour after the heating system of the train is turned on, assuming that the internal and external temperature of the vehicle is approximately -10°C before the heating system is turned on. It may be assumed that the water tank is filled and the piping and toilet system were freeze drained.[2]

2b. Whenever the internal and external temperature of the vehicle is approximately -10°C, then the toilet system shall be able to operate within 1 hour after the heating system of the train is turned on. For the purposes of testing, it may be assumed that the water tank is filled and the piping and toilet system were freeze drained.

[1] Source: SNG Spec Early draft, section 6.
[2] Source: EuroSpec *Specification for Toilets of Railway Vehicles,* 1st edition, April 2012.

Again, the tendency to state the object first means you miss the opportunity to properly emphasize (a) the strength of the imperative and (b) the proper conditions under which the requirement is to be assessed as fulfilled or failed. This last point is crucial.

3a In case of a turned off heating system, the toilet shall withstand an external temperature of -10°C for a continuous period of 12 hours without any damage , assuming that the internal temperature of vehicles was approximately +20°C before the heating system was turned off.[3]

3b If the heating system is turned off and the internal temperature of vehicles is approximately +20°C when the heating system is turned off, the toilet shall withstand an external temperature of -10°C for a continuous period of 12 hours without damage.

In the alternative form 3b, it is much easier to investigate the proper conditions under which the requirement is to be assessed as fulfilled or failed. We shall show the sentences 1 -3 are all of the conditional format in chapter (4), (5) and (6), and this allows us to show that (1 -3) are to be assessed in exactly the same manner.

4a Dedicated personnel shall be able to set the HVAC system to a "cleaning" or "anti-freeze" mode during which the heating/ventilation system regulates to an interior temperature of 5°C and the cooling system is blocked.[4]

4b Dedicated personnel shall be able to set the HVAC system to a "cleaning" or "anti-freeze". In either mode, the cooling system shall be blocked, and the heating/ventilation system shall regulate the interior temperature to 5°C exactly.

Apart from some merely aesthetic changes, there is a really important adaptation of 4a. In 4b we clearly disambiguate when we say that the following conditions (the cooling system shall be blocked, and the heating/ventilation system shall regulate the interior temperature to 5°C exactly) shall hold "[i]n either mode…". If we do not explicitly disambiguate, we risk the supplier assuming that the following conditions holds of only the latter anti-freeze mode That is to say, the reader could take the specification to mean that the heating/ventilation system shall regulate the interior temperature to 5 °C, and the cooling system is to be blocked only when the HVAC system is in "anti-freeze" mode. This is an error, but even if the author intended the sentence to be interpreted as a specifying conditions on only the anti-freeze mode, the sentence is ambiguous the other way (as exemplified by my misunderstanding). To avoid ambiguity, we should properly index the requirements to their appropriate conditions. That is to say, where we specify any modification or refinement of an object or a condition, we should ensure that the modification is clearly connected to the appropriate object or condition. This principle has a wide application, for instance consider the following example of an introductory paragraph which precedes the statement of a requirement.

5a To realise this halting time of 0.6 minutes in the current NS departure procedure (see B6.8), taking into account the time the train driver and the train guard need for their actions, the following requirements specify boarding and alighting times of the passengers, opening and closing times of doors and steps, door distribution and door width.[5]

5b Taking into account the time the train driver and the train guard need for their actions, the following requirements regarding (i) boarding and alighting times of the passengers (ii) the opening and closing times of the doors and (iii) the dimensions of the (passengers) entrance ways to the train, are specified to help realize this halting

[3] Source: EuroSpec *Specification for Toilets of Railway Vehicles,* 1st edition, April 2012.
[4] Source: SNG Spec Early draft, section 6.
[5] Source: SNG Spec Early draft, section 4.

time of 0.6 minutes in accordance with the current NS departure procedure (see B6.8)

Again, apart from aesthetic changes, the order of the sentence is deliberately inverted. Since, this sentence is supposed to serve as a statement of purpose, you are better off to begin by clearly introducing what kind of specifications you are about to provide, and for what objects, before you indicate what purpose these provisions serve. This is especially true, where you have to state a long list or detailed description of these provisions, as the reader is likely to get lost along the way. By specifying the purpose of your provisions toward the end of their description you helpfully compound the interpretation.

3.1 The moral

The kinds of errors made in these examples can be easily avoided by adopting some simple conventions regarding sentence structures. More pertinently, if we appropriately structure our sentences, we can also rigorously define the conditions under which the requirements have been met, and as such when they have been failed. This will help in determining the conditions of compliance that any supplier has to meet. We now examine the notions of sentence structures, and recommend adopting particular conventions regarding how to structure these sentences.

4 Translation and structuring

The purpose of this chapter is to introduce the symbolic language, we intend to use throughout the document. Ultimately you should be able to translate between natural language and the formal language as easily as breathing. So don't be afraid of the language's formal aspect, you will eventually see that it's a very simple language.

4.1 The point

It is helpful when writing to "translate" your natural language sentence into the formal language so that you know how to properly clarify your intent. For instance, If your sentence is a conditional sentence, better to state it as a clear (if_then_) sentence than rely on the ambiguities of a more natural phrasing. Clarify the structure for yourself, and you simultaneously clarify it for everyone else. This is perfectly natural, every writer does something analogous. In this document, we will provide you with the tools to perform a formal analysis of natural language so as to facilitate the easy assessment of requirements and a method for breaking down complex requirements into simple requirements.

Our method of formal analysis requires that we use a formal language. So we now introduce the alphabet of this language, and briefly explain their role in our system. The details will be elaborated further in subsequent chapters.

Like natural language, our formal language will enable us to talk of things, and ascribe properties to them. In addition we shall be able to describe the relationships that hold between things. In short, our formal language should be as expressive as natural language, but somewhat simply and less ambiguous!

- We translate names in natural as lower case letters (a – l). We call these constants.
- We translate properties or relations described in natural language as capital letters (A – Q). We call these predicates.
- We reserve (m, n) for numeric notation.
- We also have variables in our formal language. The variables (x, y, z, X, Y, Z) are used for object and predicate variables. The lower case (x, y, z) are variables used to describe objects, or entities, and the capitals (X,Y,Z) are used to describe properties or relations.
- In addition α, and β serve as sentence variables. Sentences are minimally at least one ascription of a property to one individual. We call this the *predicate-object structure*.

Consider the following translation manual.

1. The set of objects in the universe of discourse := {a, b, e}
2. The set of properties in the universe of discourse := {F, A}
3. F := "…is Fat"
4. A := "…is Alive"
5. a := Adam
6. b := Bill
7. e := Elvis.

Consequently, "Elvis is alive" is simply written "Ae" because we are only ascribing a property to Elvis, namely the property of being alive. If you forget the translation manual you are likely to make Elvis into a predicate, "(There is an x)(Ex and Ax)", or into a variable, (For all e)Ae". Both alternatives are silly, since the latter commits you to the existence of a plurality of "Elvises", and the former commits us to the view that being Elvis is a contingent property of thing, anything, you, me or trains. We could all be Elvis at the weekends. *Remember the predicate-object structure and ensure that all your atomic sentences conform to this structure*.

Now you might consider that we should be able to evaluate any sentence involving Elvis, more particularly, since we know that Elvis is dead (i.e. not alive) we know that it is not the case "Ae". This is exactly right, and we show how this insight is at the core of writing accurate requirements. If you can determine when and why a sentence is true or false, you can determine whether a requirement has been met. Since Elvis is dead we write Not(Ae), in formal language.

This serves two purposes, economy and clarity. Ultimately, we will show that all complex sentences can be broken to into segments of the form $X(x_1 \ldots x_n)$ where we see that some property X has be ascribed to some set of individuals, where *n* is greater than or equal to 1. The idea is that we should be easily able to determine whether such a property ascription is accurate, and so whether $X(x_1 \ldots x_n)$ is true or false. Our formal system assumes any admissable sentence can be evaluated as true or false, and so (1) we rigorously define the rules for constructing correct sentences and (2) the rules for evaluating any correctly constructed sentence.

4.2 The language: combining alphabet with operations

In the last section we merely elaborated the alphabet of our language and discussed simple sentences, but there is more to our language than constants, variables and predicates. In particular we need operational connectives, to translate words such as (and, if, or… etc.) these words are used in our language to relate sentences, and join small sentences into bigger sentences. We define this a bit more rigorously.

Let α and β be variables for admissible sentences. Any admissible sentence α is composed of elements. These elements are either operational connectives, names or predicates. We distinguish between (1) atomic sentences and (2) complex sentences.

> **Atomic :=** An atomic sentence is any sentence of the form $X(x_1 \ldots x_n)$ where *n* is less than or equal to 1. Atomic sentences are admissible by definition.
>
> **Complex :=** Any sentence composed by means of simple operations upon Atomic sentences. A complex sentence is admissible just when it is composed by means of a sequence of *n*-steps, with any of the six operational connectives below.

Let α and β be variables for any admissible sentence-type. The rules of composition are as follows.

1. Any sentence of the form - **Not**(α) is an admissible sentence.
2. Any sentence of the form - ((α) **and** (β)) is an admissible sentence.
3. Any sentence of the form - (**If** (α) **then** (β)) is an admissible sentence.
4. Any sentence of the form - ((α) **or** (β)) is an admissible sentence.
5. Any sentence of the form - (**There is an x**(α[x])) is an admissible sentence.
6. Any sentence of the form - (**For all x**(β[x])) is an admissible sentence.

These rules allow for any admissible combination. So an imaginative use of these structures allows us to state the complex conditions in an easy to assess manner because we shall develop rules to evaluate any sentence of the six types. This language allows for almost a complete formalization of

natural language, but for most purposes we will only need a semi-formal approach – just enough formality to clarify sentence structures.

Let □ be a variable for some operational connective.

Any sentence composed with □ will depend on certain rules of construction regarding how to use □, these rules are often called the truth-functional rules for □. This is not too important, the key point is that we can define rules for when it is appropriate to use □, and when it is inappropriate. It might seem that in natural language we rarely use the □ operation, as such most of our sentences do not look like □(α) or (α □ β). This document aims to show you that most of the sentences used in natural language can be understood to have a singular form, and that form is determined by the operational connective.

There are just six operational connectives in our language, we name and represent them.

- Negation: **Not**(α)
- Conjunction: α **and** β
- Disjunction: α **or** β
- Implication: **if** α **then** β
- Existential quantifier: **There is some** x (α(x))
- Universal quantifier: **For all** x (α(x))

Each sentence is characterised by one of the above operational connectives. Allow that □ is one of the above operations. *Can you think of how you might define the rules for when its appropriate to use □?* Suppose that,

□ := and,

then you should naturally think that we can only use α □ β whenever it is appropriate to assert that α while it is also appropriate to assert that β. We will discuss such considerations further in chapter (6). However we note now that each sentence-form/type is schematic, and has an infinite number of instances. For example, consider the cases, where the universe of discourse is defined as above.

1. **Negation:** Not(Ae)
2. **Conjunction** (Ae and Fb)
3. **Disjunction**: (Ae or Fa)
4. **Implication**: (If Fe then Fa)
5. **Existential**: There is some x (Fx and Ax)
6. **Universal**: For all x (Ax)

NB: The set of constants (a….n) can in our formal language, be used to define more things than just names. For instance if we have a unique or definite description such as "The King" we can either treat this description as a name, and translate as such, or we can introduce a variable and say there exists an x, such that x is the King, and nothing else is. In other words, we have two options

- The King:= e
- The King := There is exactly one x (Kx)

Considerations of context should be applied in determining how you use these options. We should note that there tends to be some confusion over how to interpret the quantifiers [i.e; There is some x (α(x)), and For all x (α(x)),] when α is a sentence variable. We want to distinguish between the level of quantification, in the example above, we have quantified over the universe of discourse, but we can also abstract one level and quantify over all the objects in a sentence. We represent this as follows:

There is an x (α(x))

Simply means

There is a thing x, mentioned in the sentence α, and x satisfies all the properties ascribed to it in the sentence α.

Whereas

For all x (α(x))

means

For all x, mentioned in the sentence α, x satisfies the properties ascribed to it in the sentence α.

In contrast, when we say that

For all x, (Fx),

We simply mean that everything is such that it satisfied the property of F (i.e. is fat), and similarly, for the claim ˙There is an x, (Fx). We will discuss the rules which govern the operational connectives in chapter (6). For now we wish only to emphasise the structural properties of the language. An understanding of these sentence structures is vital for the application of our proposed method .In the next chapter we shall flesh out the details of our language's syntax.

5 The syntax

We now turn to the grammar of our language. As above, we make the following idealisation all sentences are of one of two types. The goal of this chapter is to make clear the procedures by which (a) atomic sentences are constructed by means of ascribing properties to objects and (b) how to construct complex sentences by combining atomic sentences by means of admissible operations.

1. **Atomic sentences** are considered to be unit-sentences of which more complex sentences are composed. They are by definition simple, e.g. "The train set is in mode "Ready", which we can translate as "Rt" if we assume some appropriate mapping between the formal language, and natural language.
2. **Complex sentences** are composed of a number of sub-sentences, which are either themselves complex or atomic, e.g. ((The train set is in mode "Ready") and (the train set is in the station or not in the station)) which we can translate as - (Rt and (St or Not(St)))

The rough analogy is between the chemical composition of molecules from atoms. The atomic sentences are bound together by means of a compositional operations into increasingly complex sentences. So analogous to the manner in which a chemist can discern the atomic structure of a molecule, we too should be able to the discern the atomic structure of any sentence if we follow the rules of construction as describe below.

5.1 The syntax: atomic sentences

So called simple sentences often contain more information then is strictly needed. Also, the simplicity of a sentence is determined by the reader and as such we should not presuppose that our sentences are intuitively simple when we write requirements. To avoid such assumptions we define the notion of a simple sentence in terms of the complexity of their construction. Sentences like buildings are composed of bricks, or basic elements. The object of a sentence is the thing which we are talking about, the properties of that object are described by the predicate we ascribe to that object.

> **Atomic :=** Any straightforward declarative statement of the form "a is F" or "a stands in the relation F to b", where "a" is a naming term of some object and "F" is some descriptive, or relational property. In other words F is a property that is either ascribed to one or many individuals. The structural format is know as a function/argument format, any atomic sentence can be written in the form $X(x...x_n)$, where X is a function ascribing a property to the argument object(s) $x...x_n$. Atomic sentences can involve a number of arguments, but must only have one function. Sentences of the form F(a)(b) are a function ascribing a relationship F to the ordered pair <a, b>. The notion of function/argument structure may help clear up the idea for the mathematically minded reader. If you do not find it clear, then continue to consider atomic sentences as simple declarative sentences involving some single property ascription to a number of individuals, also called the predicate-object structure.

There are two clarifications.

If F is defined as a simple (monadic) descriptive property such as F = (...is Fat) then the object of the sentence *a* is said to be fat. Example; the sentence "*The train sets* can couple" can be read as a sentence of the form (a is F).

However, if F is a (polyadic) relational property such as F = (...drives...) then we have cases such as (a drives b), or more unnaturally (a and b stand in the relation of driving). Example; The sentence "The *active train set* shall couple with *the passive train* set" can be read as

(a shall couple with b)

Note that atomic (polyadic) sentences can speak of any number of objects. They are atomic in virtue of the fact that only one property relates these objects. By insisting on this restriction all our atomic sentences will be simple and informative.

If we allow that α and β are variables for atomic sentences, we can break down the structure further. Since all atomic sentences refer to objects and their properties. Every object which exists has properties, so we know that whenever we specify an object we can do so by describing its unique properties and saying it exists. We can also name any object we conceive of, in any relation whatsoever, so any atomic sentence will either have the form:

F(a) = (The train set$_{(a)}$ shall have doors$_{(F)}$)

Which is simply to say, that *a* is one of the things which satisfies the property of F. Castles have doors too, but there is no need to mention them. Keep the sentences simple, and relevant.

Or perhaps the sentence has an undefined object.

F(a)(y) = (The doors of the train set$_{(a)}$ will be easily accessible by$_{(F)}$ somehow$_{(y)}$[6])

Which is simply to say that *a* stands in the relation F, to something y. If we find out the name of the thing (y) is (b), we know that F(a)(b), or that a is F to b. We call (y) a free variable because we haven't quantified over it, we're not saying that it exists, or that all (y) are F-related to (a). When we use a free variable, we tend to be deliberately vague or "hand wavy". It is far better to avoid the use of free variables. Consider:

The doors to the train will be easily accessible somehow.

Against,

There is some thing-method, y such that the doors to the train will be easily accessible by (y).

…and…

The doors to the train will be easily accessible via the platform.

Try to tightly define the object terms. You have not described an entity unless you can state its identity conditions, that is, the conditions anything would have to satisfy to be considered identical with that thing you intend to describe. You have not described a safety feature unless you can properly individuate it from the control system, similarly, you have not described an elephant unless you can distinguish it from a rhino. We describe entities as objects with certain properties. Linguistically, we have names for objects, and predicates for properties. These are our building blocks. If the concepts are unclear continue to read, otherwise, you may skip to subsection 5.2.

5.1.1 Further examples

If we want to say something more complicated we can define the relation F to involve more objects such that

F(a)(b)(c)

Which is just to say *a* stands in F, to *b*, who in turn is F-related to *c*. For instance, consider the two (perhaps) ambiguous examples.

Adam is the father of the father of Cecil, who is Bob.

NS is the employer of the employer of the supplier, that is, Lloyds.

[6] Strictly speaking, this sentence is needlessly ambiguous, it would be better to replace the notion "easily" with some more exact property, but lets stick to one problem at a time.

In general you shouldn't need more than simple binary-relations (i.e. relations between two objects).To create more detailed requirements it is better to state them separately as distinct sentences, then cramming them together under one complicated *n*-ary relation.

We can clarify the structure we want by breaking the sentence down in terms of its atomic constituents, and changing the relation to a binary.

((Adam is the father of Bob), and (Bob is the father of Cecil))

Terminologically, we say that all relations have an arity, and the arity of the relation is determined by the number of individuals it relates. Typically, when we have a complex relation of an arity > 2 we have unclarity. For instance, romance novels typically rely on the premises that the arity of the love-relation is imagined to be more than two. We note the arity of a relation by subscripts. It would be boring if

((a loves$_2$ b) and (b loves$_2$ a))

((a is coupled$_2$ with b) and (b is coupled$_2$ with a))

but dramatic if

((a loves$_4$ the lover of c, b who also loves d) and (d loves$_4$ the lover of b, c who also loves a))

((a drives$_4$ the train set belonging to c, d that is repaired by the mechanic b, who is responsible for the safety standards of the train sets) and (the mechanic b, who is responsible for the safety standards of the train sets is jealous$_4$ of the driver a, of the train set d, belonging to c)

Generally, avoid complex relations when writing requirements. Break up the sentences, and introduce names where possible.

Warning! Not all short descriptive sentences are atomic, sometimes there can be more information contained in a sentence than is advisable, as this adds to confusion.

For instance,

(5a) Tony Soprano is a family man, Mafioso.

(5a*) The time in which the doors and the step of each passenger entry make the complete opening movement shall be <= 4.5 seconds[7]

The sentence (5a) is not atomic. It speaks of some object i.e. Tony Soprano, but ascribes two properties to the man, namely that he is both a family man and a Mafioso. Worse still, these properties could be contradictory or conflated, e.g. is Tony loyal to his family i.e his wife and children, or is he loyal to the "the Family"? This unfortunate ambiguity is good for the script writers of *the Sopranos,* but we can't afford such vagueness in the writing of a specification. Similarly, in (5a*) we cannot be sure whether the requirement is intended to mean that the doorway and the step undergo an opening movement. Or if just the doorway opens while the stairway is static. Better to have

(5b) (Tony Soprano is a family man), and (Tony Soprano is a Mafioso).

(5b*) (The time in which the entranceway of each passenger door takes to complete the opening movement shall be <= 4.5 seconds) and (if (the design is such that the stairway in each passenger door is able to undergo an opening movement), then(it shall occur simultaneously with the opening of the doorway and take <=4.5 seconds)).

[7] Source: section 4 early SNG draft.

The choice to predicate only one property on individual(s) per sentence allows for increased clarity, and avoids the potential for needless ambiguity. Strictly speaking, we have multiple atomic sentences in our examples. The first says that the "The doors shall open in <= 4.5 seconds" and the second says that "the steps shall open in <= 4.5." Once we have identified each atomic requirement, we can refine them by describing the kinds of additional information or conditions under which our requirements have to be satisfied e.g. (5b*). What is crucial is that we first identify the atomic sentences of any requirement, for if we can determine which are the minimally desired constituents of a sentence, we can learn to clearly write up the sentences. How would you avoid the ambiguity of "it" in 5b*? Remember that in this context "it" serves as a name, is it clear what the term names? How would you using quantifiers specify that the object of the sentence is an opening movement?

5.2 The syntax: complex sentences

We build a complex sentence out of atomic components. All atomic sentences are admissible sentences and any complex sentence composed by means of our simple operations (1 – 6 below) on atomics is an admissible sentence. We define the class of increasingly complex admissible sentences recursively, as one which includes all and only admissible sentence combinations

By analogy, consider the following case. If (2+2) is an atomic, it is by definition admissible, and if "+" is an admissible mathematical operation then ((2+2) + (2+2)) is admissible sentence. The idea is that similar definitions can be made for when certain kinds of atomic sentences are legitimately composed. We first elaborate the kinds of compositional structures we shall consider in our language, and then show how they are utilised.

Complex := Any sentence composed by means of simple operations upon Atomic sentences.

Let α and β be variables for any admissible sentence-type. The rules of composition are as follows.

1. Any sentence of the form - **Not**(α) is an admissible sentence.
2. Any sentence of the form - ((α) **and** (β)) is an admissible sentence.
3. Any sentence of the form - (**If** (α) **then** (β)) is an admissible sentence.
4. Any sentence of the form - ((α) **or** (β)) is an admissible sentence.
5. Any sentence of the form - (**There is an x**(α[x])) is an admissible sentence.
6. Any sentence of the form - (**For all x**(β[x])) is an admissible sentence.

These rules allow for any admissible combination. So an imaginative use of these structures allows us to state the complex conditions in an easy to assess manner. This language allows for almost a complete formalization of natural language, but for most purposes we will only need a semi-formal approach – just enough formality to clarify sentence structures.

5.2.1 Examples of semi-formal translation

We now consider a few examples of complex sentence constructions.

7a Its not the case that$_{Not}$ Tony Soprano$_{(t)}$ is just a Mafioso$_{(M)}$ or a Family man$_{(F)}$, he's both.

7b The doorway entrance$_{(d)}$ shall not be less than XXXX wide$_{<W}$ and XXXX high$_{<H}$.

There are a number of ways to represent this complex claim, so it is important to pay attention to the rules of composition, and present information carefully. Two candidate interpretations of (7a) are

7a* **Not**((Tony$_{(t)}$ is a Mafioso$_{(M)}$ or Tony$_{(t)}$ is a Family man$_{(F)}$) and (Tony$_{(t)}$ is a Mafioso$_{(M)}$ and Tony$_{(t)}$ is a Family man$_{(F)}$))

7a** (Not(Tony is a Mafioso or Tony is a Family man)) **and** (Tony is a Mafioso and Tony is a Family man)

In (7a*) the negation takes the conjunction (P &Q), and in (7a**) the negation only takes the disjunction (P or Q). But if we treat 7a as being analysed as (7a*), then we know that Tony is neither Mafioso nor Family man, and all we have done is to say two things Tony isn't. This is not very informative, for its also true that Tony isn't a light bulb but that's rarely interesting enough to mention! *Whenever you state a requirement its best to contribute relevant information.* Negative information is rarely sufficient. As such, its better to treat (7a) as being analysed in terms of (7a**) because in this case we now know that Tony is both a Mafioso and a Family man. As a rule of thumb we should aim to be as informative as is relevant, and assume our others do the same.

Similar considerations apply to the analysis of (7b) We have two candidate readings

7b* Not((the doorway entrance$_{(d)}$ shall be less than XXXX wide$_{(<W)}$) and (the door way entrance shall be less than XXXX high$_{(<H)}$))

7b** (Not(the doorway entrance shall be less than XXXX wide)) and (the doorway entrance shall be less than XXXX high)

For both (7a*) and (7b*) we say that the form of the sentence is a negation, whereas for (7a**) and (7b**) the form of the sentence is a conjunction. This is because the main operation (or connective) is in the latter case a conjunction, and in the former a negation. This is easily observed by determining which connective has the broadest scope.

5.2.2 Avoiding ambiguity with quantifiers

The brackets we used above are a notational convention to help define the scope of each operation. As you can imagine, it can get quite messy if we always properly define the form of a sentence, consider the following example and translation.

(If, (for all x (P[x])), then ((there is some y,(S[x, y])) and (Not(a = y))))

In words

If any train set $_{(x)}$ is parked $_{(P)}$, then there is some building $_{(y)}$ in which it $_{(x)}$ is stored $_{(S),}$ and that building $_{(y)}$ is not the headquarters of NS $_{(a).}$

When it is clear from context, we often leave off some of the outer brackets. So a clearer rewrite of the above sentence in its abstract form is as follows:

If, for all x (Px), then ((there is some y (S [x, y])) and (Not(a = y)))

Here, we haven't put brackets around the antecedent, because there is no ambiguity as to what the condition is. However, we needed the brackets to disambiguate the consequent, because it is more complex. The building "y" is mentioned twice, once when it is introduced, and again when it is referred to as "that building." It is crucial that we allow the scope of the existential quantifier to "range" over the entire sentence. If we did not, then we could not "refer backwards" to the previously described entity, and so the value of the term "that building" would be undetermined. To avoid ambiguity, one shall properly index the requirements to their appropriate conditions. That is to say, where we specify any modification or refinement of an object or a condition, we should ensure that the modification is clearly connected to the appropriate object. When using quantifiers it is crucial to make sure that whenever you are talking about some variable object or entity, that you are aware of the scope. For instance, consider

(9a) The design shall be basically be as specified in Appendix XXX and the ENSi23241 document. Those conditions will be assumed in all the testing of the design.

Semi-formally,

(9b) There is an x, There is a y, and There is another y, such that (x is a Design and y is a Specification and y is an Appendix) and (x will be basically be designed according to the conditions in y. Those conditions in y, will be assumed in the testing of x.

There is a clear ambiguity here between the conditions specified in Appendix XXX, those specified in ENSi2341 and the conditions of both documents taken together. This problem is very easy to resolve by adding a qualifying remark, but the problem would never happen if the authors are aware of the scope considerations. As such, it's a good idea when referring to some previously discussed specific object or thing, that you always check that there is no other candidate thing that a reader might think you were speaking about. In other words we want to preserve the idea, that whenever we introduce an object to be discussed, we do not introduce too many object of the same category (i.e we should never have one variable "y" for a two potential objects. However, it is perfectly fine to introduce multiple objects . For instance

There is an x, and There is a y,(x is a man and y is a train) and (x loves y).

There is a man who loves a train.

The objects of the above sentence are two but the subject is clear. The objects are simply related to one another. It is preferable when using any relational predicate, that we are sure that the reader can understand which objects the relation speaks of.

Rules regarding atomic sentences

- If you are not sure yet what the requirement should express exactly, define its characteristics in atomic sentences first.
- When incorporated in a complex requirement, simple sentences should remain clearly distinguishable which can be checked by bracketing them.
- Try to stick to binary relations.
- Remember the predicate-object structure.

Rules regarding complex sentences

- Whenever you state a requirement it's best to contribute relevant information. Negative information is rarely sufficient.
- Whenever you negate two or more claims be sure that the scope of the negation is clear. Whenever you introduce two or more existential claims concerning the same object type e.g. two documents, be sure that any subsequent mention of a document is not ambiguous between the two previously mentioned or a third new document under discussion.
- In short, pay attention to scope, this is only possible if you have first determined which are the atomic elements of the sentence. So individuate the atomic elements of any sentence!
- To avoid ambiguity, one shall properly index the requirements to their appropriate conditions. That is to say, where we specify any modification or refinement of an object or a condition, we should ensure that the modification is clearly connected to the appropriate object.

6 The semantics

In the previous chapter we considered issues of syntax, now we turn to the semantics. The semantics of a language determine under what conditions an utterance or sentence in the language can be said to be true. The goal of this chapter will be to show there are six simple procedures by which you can evaluate any admissible sentence.

For instance, the sentence "All Triangles have three sides" is true only when it is a fact that all triangles do indeed have three sides. This seems simple, but the key insight is that once we observe this kind of pattern we can begin to abstract sentence schemas which we know how to evaluate.

From

All triangles have three sides

We get

For every x (if x is a T then x has 3).

Into which we can substitute for example

For every x (If x is a Christians then x is killed by Lions.

…and intuitively both are true, just when every x satisfies the properties ascribed. We know this just by considering the definition for "all".

The **crucial idea** of such a regimented semantics is that it allows us to determine what conditions need to be true in order for any admissible sentence to be true. This in turn allows us to break down every complex requirement to assess just when it has been fulfilled. Once you have grasped the definitions in this chapter we will turn to the next chapter and elaborate a step by step procedure for identifying if and when a supplier has fulfilled their requirements.

In this chapter we elaborate the rules to aid in this process of evaluation. We return to the six admissible operations, as mentioned in the previous chapter, and provide definitions and methods of application for each operation. We first give a tabular summary of the key points.

Rule	*Negation*	*Conjunction*	*Conditional*	*Disjunction*	*Existential*	*Universal*
Syntactic relation	Not(….)	(….and….)	(If…then…)	(….or….)	There is x(x is….)	For all x(x is ….)
Semantic evaluation	Not(…) is true iff[8] (…) is false. Otherwise it is false.	(…and…) is true iff neither conjunct is false. Otherwise it is false.	(if…then…) is true iff either the antecedent is false, or the consequent is true. Otherwise it is false	(….or…) is true iff either disjunct is true. Otherwise it is false	There is an x (x is…) is true iff there is an element of the universe of discourse such that the entity satisfies the properties ascribed to x.	For all x (x is …) is true iff every possible individual in the universe of discourse satisfies the properties ascribed to x.

[8] We use the abbreviation "iff" as short for the phrase ""if and only if".

The above summary is intended for use as reference material. It is not sufficient alone, but must be used in conjunction with the rest of this chapter in which we introduce the rationale for each of the rules.

6.1 Negation or "Not…"

Negation is a unary operator which means it can only be applied to one admissible sentence at a time. That is to say the scope of the negation operation is a single admissible sentence.

Not(a is F)

The train set is not operational.

Its not the case that the train set is operational.

Not((a is F) and (a is G))

Its not the case that the train is in Utrecht and operational.

As a statement of a requirement:

A negative requirement (P) is to be considered fulfilled if and only if NS have observed that it is not the case that (P).

Otherwise the requirement has not been met.

Tips for using negation

Where possible don't use negation. If you have try to restate the claim positively.

For example, the requirement

"The interval for any preventive maintenance shall not be less than 4 months"[9]

should be rewritten as

"The interval between any preventive maintenance shall be bigger than 4 months."

However, you should be aware of how negation interacts with other operational connectives. We will treat this issue throughout the elaboration of the five other connectives.

6.2 Conjunction or "…and…"

Conjunction is a binary operation which means it can only be applied to two Basic sentences at a time.

(a is F) and (b is G)

The train set is operational, and the timetable is accurate.

The train set will be partially yellow, but also, it will be partially grey.

As a statement of a requirement:

Any conjunctive specification (P and Q) is to be considered fulfilled if and only if NS have observed that both that (P) and that (Q). Otherwise the requirement has not been met.

[9] Source: EuroSpec *Specification for Toilets of Railway Vehicles,* 1st edition, April 2012.

By examining this definition we can see that the formula Not(P and Q) is true just when the formula (P and Q) is not true, i.e when the conjunctive specification has failed. So we know that from Not(P and Q) it follows that (Not(P) or Not(Q)) or (Not(P) and Not(Q)).

Tips for using a conjunction

We express conjunction with many words other than "and", including "but," "moreover," "however," "although", and "even though". In English these expressions sharply contrast the two conjuncts, saying in effect "if you believe the first conjunct, then you will be surprised by the second." But they still assert a conjunction.

- The train sets shall be operational by 2013, even though thirteen is an unlucky number.

This sounds odd, but it makes more sense when you think of “even though” as indicating a grudging acknowledgement of the fact that 13 is an unlucky number. It serves the same function no matter what the nature of the second conjunct.

Sometimes "and" does not join atomic sentences into a complex sentence . Sometimes it simply joins nouns:

(a) "The functioning of the control system and the diagnostic system is interdependent“.

This cannot be paraphrased as

(b) "The functioning of the control system is interdependent and the functioning of the diagnostic system is interdependent",

for that does not assert that they are interdependent upon each other.

Sometimes "and" joins adjectives:

(c) "The entrance way of the train set shall be easily accessible, effectively XXX wide and optimally positioned relative to the station platform."

This, however, can be paraphrased,

(d) “The entrance way of the train set shall be easily accessible. The entrance way to the train set shall be XXX wide. The entrance way of the train set shall be optimally positioned relative to the platform.”

Typically, where there is a long conjunction, break it up into simply stated atomic sentences. In this manner each requirement can be observed to be failed or fulfilled individually. Atomic sentences are easily checked for compliance.

6.3 Conditional or “If…then…”

The conditional operation is a binary operation. Any conditional sentence asserts an order of dependence between its constituents. The antecedent (P) necessarily ensures the truth of the consequent (Q).

(e) If the train set is operational, then the <<control current key>> has been entered.

(f) If P then Q

This is equivalent to saying that (Q) is a minimally necessary condition for (P) to be true, which means that; if Not(Q), then Not(P). Intuitively, the truth of the following sentence “ if x is a cat, then x is a mammal” ensures that “if x is not a mammal, then x is not a cat.” As such, the conditional format will be crucial for writing up general requirements. It allows us to state the condition, that we want met, modified, refined or qualified. As a statement of a requirement:

Any conditional specification is to be considered fulfilled if and only if NS has observed that either **(i)** *Not(P) is true, or* **(ii)** *(Q) and (P) are true.*

Otherwise the requirement has been failed. In other words, from Not(if P then Q) we know that it must follow that (P) is true and Q is false.

Tips for using the conditional

The conditional appears in many forms in the English language, for example the following constructions mean exactly the same thing: "if p, then q", "if p, q", "p implies q", "p entails q", "p therefore q", "p hence q", "q if p", "q provided p", "p which means, q" "q follows from p", "p is the sufficient condition of q", "q is the necessary condition of p" "p depends on q".

The least intuitive is "p only if q", we will discuss this last case below.

Ambiguity and Vacuity

Be careful. Some nearly synonymous expressions, like "q because p", "q since p", "because p, q", "since p, q", and even some instances of "p therefore q", are not genuine cases, or at least not merely cases, of the conditional. They may seem so because they make p into a condition or reason for q. But in "if p, then q" we are non-committal about the truth of p, whereas most speakers who assert "q because p" and its variants are asserting the truth of p. To capture this aspect of the proposition's meaning, use conjunction, "q and p". To capture the implication claim as well, use both conjunction and material implication, "Possibly p and (if p then q)". For most purposes, in requirement writing, we shall always assume that the truth of the antecedent p is possible, so by writing e.g.

> "Since the train set is the mode "Ready", the train set will transition to the mode "operate" when the control current key is entered."

We presuppose that the train set is in the mode "ready". However, when writing a conditional requirement we need not do so. For instance, if we are specifying a conditional requirement on the basis of some optional condition then, the supplier does not need to meet the requirement if they do not provide the option. So where, the train has no golden toilets, the following specification is fulfilled.

(g) If the toilets are made of gold, they must be accessible only to staff.

(h) If (p) then (q).

We say, in such cases, that such a condition is fulfilled vacuously because there are no golden toilets, so they couldn't be accessible to staff at all. This means whenever we write a requirement of the conditional format we should ensure that the antecedent can be fulfilled. If we wish to stress this fact, we should write.

(i) Because the toilets are to be made of gold, they must be accessible only to staff.

Or more naturally

(j) The toilets must be accessible only to staff because the toilets are made of gold.

Necessity and Sufficiency

Understanding the conditional relationship allows us to understand the necessary and sufficient conditions of any requirement. We say that p is a sufficient condition of q when p's truth guarantees q's truth. By contrast, q is a necessary condition of p when q's falsehood guarantees p's falsehood. In the ordinary conditional, "if p then q", the antecedent p is a sufficient condition of the consequent q, and the consequent q is a necessary condition of the antecedent p.

Satisfy yourself of this by reflecting on an example:

(k) "If Socks is a cat, then Socks is a mammal." Being a cat is a sufficient condition of being a mammal. Being a mammal is a necessary condition of being a cat.

However, there may be many sufficient conditions for being a mammal. For instance, if we substitute “Socks is a cat” for “Socks is a dog”, the value of the sentence doesn’t change we have still asserted a necessary condition for being Socks.

The fact that material implication expresses necessary and sufficient conditions in this way can be a great help in translation. Ask yourself about a difficult English sentence: what is being asserted to be the sufficient condition of what here? What is being asserted to be the necessary condition of what here? When you find the sufficient condition, make it the antecedent. When you find the necessary condition, make it the consequent. If p is both necessary and sufficient for q, then we must say "p if and only if q".

Only if.

We translate "p only if q" as "if p then q". This is surprising to many people because "if" usually cues the antecedent. Rather than say that "if" sometimes cues the consequent, it is better to say instead that "only if" differs from "if", and "only if" cues the consequent. If you understand necessary and sufficient conditions, this translation should make more sense: "p only if q" clearly asserts that q is a necessary condition of p. The necessary condition of something is the consequent of that thing in a conditional. For example

(l) Something is a train set only if it has wheels.

Which can be understood to say

(m) If something is a train set, then necessarily it has wheels.

But as we’ve seen above, this is equivalent with

(n) If something is a train set then it has wheels.

Which is just the standard conditional operation. By making our conditional operation so strong it means that we lose any ambiguity whatsoever. You should be aware of this as a writer and a reader. If you don’t want to insist that something is the necessary consequent of a conditional, then you should suitably qualify your claim with “probable” or “optional”.

6.4 Disjunction i.e. “...or...”

The disjunctive operation is a binary operator. Any disjunctive sentence presents an option between two disjuncts. A disjunction can be exclusive or inclusive, by which we mean that

(a is F) or_i (b is G)

can be considered true where only one of the disjuncts is true, but also when both disjuncts are true. In the first case, we mean the disjunction to be an exclusive disjoin between two options. In the second case, a disjunction can also be considered true where both disjuncts are true, as in the case of conjunction. This is reasonable because it means, that the truth of any conjunction will entail the truth of any disjunction of the same components. For instance, its natural to say

(o) John and Bill will come to the party

entails

(p) John or Bill will come to the party

Which means, that If (John and Bill will come to the party) then (John or Bill will come to party).

However, it would be wrong to say that

(q) Luke will join the dark side, or the light side.

entails

(r) Luke will join the dark side and the light side.

Which means, Not(if (Luke will join the dark side or the light side) then (Luke will join the dark side and the light side.))

As a statement of requirement disjunctions can be used in two senses. We stipulate as follows

Any (exclusive) disjunctive specification(P or Q) is to be considered fulfilled if and only if NS observes that either (P) or (Q) but not both, is true. It will be failed otherwise.

Any (inclusive) disjunctive specification (P or Q) is to be considered failed if and only if NS observes that neither disjunct has been met i.e (Not(P) and Not(Q)). It will be fulfilled otherwise.

The exclusive disjunction can be used to specify a choice between two incompatible options. Whereas the inclusive disjunction can be used to present a series of options all of which can be pursued or implemented together. We can use subscripts to specify which kind of disjunction we are using if the context isn't clear.

- Assume an inclusive disjunction, then Not(P or Q) implies (Not(P) and Not(Q))
- Assume an exclusive disjunction, then Not(P or Q) implies (Not(P) and Not(Q)) or (P and Q).

Tips for using Disjunction
Sometimes "unless" should be translated as inclusive disjunction, and sometimes as exclusive disjunction. In writing requirements however it should be mostly treated as an exclusive disjunction.

For example, "The train set will be in "operation" mode unless the control current key is removed" means that the train set will be in operation mode if the control current key is not removed. In other words, saying that

(a is F) unless (b is G)

Is equivalent to saying either

(Ex) (If Not(b is G) then (a is F)) and (Not((a is F) and (b is G)))

(Inc) If Not(b is G) then (a is F)

depending on whether "unless" is interpreted as exclusive (Ex) or inclusive (Inc).

And/or
When people say "and/or", they seem to mean inclusive disjunction. For example, "The train set will be in the mode "Ready" and/or clean" means that the train will be in either "Ready" or "Clean" mode, or both.

6.5 Existential Quantifier or "There is some/a…such that…"

The existential quantifier is unary operation which can be viewed as taking single sentences as inputs and outputs a schematic sentence structure. This procedure allows for systematic generalization. So instead of continually writing "a is F and b is F, and c is F…and Not(e is F)…..*etc"* we can abstract and say

There is some x, such that (x is F).

There is a train set, and it is broken.

As a statement of requirement this allows us to introduce object-variables and specify the requirements that this object shall have. It also allows us to introduce and characterize a variable before identifying it with some particular object or component. In other words

There is some x, such that ((x is F) and (x = a)).

There is a broken train set, and it is my childhood toy.

The quantification operation is crucial to writing detailed requirements. So it is very important to understand its usage.

Any existentially quantified statement of a specification is deemed fulfilled if and only if NS has observed that there is indeed some object(s) x in the supplier's proposal/design and that they satisfy the property of being F.

The object-variable can range over any and all types of object, from chairs and tables, to relations, and structures, engines, and toilet seats. We can quantify over anything that can be named. Consider the following schema

There is an x, There is y such that if x is F, then y is G.

There is a train set and a mechanic, and if the train set is broken, then the mechanic is sad.

Now we substitute names and properties other names, or properties into the same structure

There is an (active train) set and there is a (passive train) set, if (the active train set) is able to couple, then it can couple with the passive train-set.

This sentence will be false if there is no means of defining an active or passive train in the supplier's offer because then NS cannot observe that there is any active train. The key idea is to ensure that the structure of the written sentence is in the form of an admissible sentence. If so, then any substitution of the variables will be an admissible sentence. Its natural, then to think that the failure conditions are as follows From,

Not(There is x(Fx))

we know that there is nothing in the universe of discourse that satisfies property of being or having F. Or equivalently, for all x in the universe there is no x that satisfies F.

6.6 Universal Quantifier or "For all…such that…"

The universal quantifier allows us to state general case requirements. It is also a unary operator in that it abstracts over single sentences. Instead of ((a is F) and (b is F)) and (c is F…etc) we can write

For all x such that (x is F).

All Romans are Christians.

All Christians are Catholics.

All Romans are Catholics.

This is crucial if we want to list a set of requirements for a general class of objects. So as a statement of requirements:

Any statement of a universal specification shall be considered fulfilled if and only if NS observes, that all the objects x in the supplier's proposal/design are F.

These six structures make up every permissible sentence unit in the language of our specification. They are more than sufficiently expressive to specify any conditions you care for. The failure conditions are as expected. From

> Not(For all x(Fx))

We know that, there is some x in the universe of discourse that does satisfy F, for otherwise all things in the universe of discourse would satisfy F, but this is contrary to what we know. It is important to always track the interactions between the effect of the operations on each other.

Tips for using the quantifiers

The Universe of discourse.

Remember that quantifiers implicitly refer to the universe of discourse. The universal quantifier doesn't strictly say, "for all things", but more precisely, "for all things in the universe of discourse". Similarly, the existential quantifier does not merely say, "for at least one thing", but "for at least one thing in the universe of discourse". By convention the universe of discourse is unlimited unless we stipulate otherwise. As such, all informative uses of a quantifier will be restricted such that the reader may somehow discern what the universe of discourse is. i.e. are we talking about train sets, diagnostic systems or train tracks? Read statements with universal quantifiers as if the subject were "everything whatsoever" [in the universe of discourse]. After a little practice, you'll discover how to make ordinary assertions in this unusual idiom. "Everything whatsoever, if it's a dog, it goes to heaven and if it's a train, it goes to Utrecht."

Universal quantifiers typically take conditionals. All humans are mortal: (For all x)(if Hx then Mx)". To see this, note that statements about "all" things of a certain kind (e.g. all humans) contain an implicit "if, then" structure: For all things in the universe, if they are humans, then they are mortal. To appreciate why universal quantifiers usually take conditionals, try a conjunction to see how absurd it would be. "(For all x)(Hx and Mx)" says that everything in the universe is both human and mortal — including the dogs in heaven and the trains on the track.

Sometimes we want to *limit the universe of discourse* to things with two or more properties. We indicate this limitation with conjunction. Feel free to use conjunctions in this way even if the main operation is another connective. For example, "All fat bats are mammals": "(For all x)(if (Fx and Bx) then Mx)". Here the universal quantifier still takes a conditional; but the antecedent of that conditional is a conjunction. If we want to limit the universe to things with three or more properties, we just enlarge the conjunctive string, e.g. "All old, intellectual, troubled, rabid bats are mammals": "(For all x) If (((Ox and Ix) and (Tx and Rx)) and Bx) then Mx]".

Universal quantifiers should generally not take negated conditionals, since they are equivalent to conjunctions (i.e conjunction of the antecedent and the negation of the consequent) —and you rarely want to universally generalize a conjunction. In other words, don't write

> For all x, Not(if Fx then Hx).
>
> For all train sets, it shall not be possible that if x is in mode "Ready" then x is not capable of braking.

Because, you may as well write

> For all x (Fx and Not(Hx)).
>
> For all train sets, x is in mode ready, and x can brake.

However, you are not typically able to distinguish every object in the universe of discourse by two properties. As a result you should refrain from universally negated conditionals, as they can typically be rephrased positively.

> For all x, if x is in mode ready, then x can brake.

Existential quantifiers typically take conjunctions.

"Some humans are inhumane": "(There is some x) such that (Hx and Ix)". To see this, paraphrase the original thus: some things have two properties, namely, being human and being inhumane.

Avoid existentially quantified conditionals. This is only a rule of thumb, not a rigid ban. Existentially quantified conditionals are "grammatical" in our notational language, but are almost never good translations of English language conditionals. For example, "If something falls into a black hole, then it will be lost forever," reads at first as if it takes an existential quantifier: "(There is an x)(If Bx then Lx)". But this formula is equivalent to "(There is an x)(Not(Bx) or Lx)", and "(There is an x) Not(Bx and Not(Lx))" which are not conditionals. They are far from what the speaker of the original English sentence probably meant: there is something which is either not in a black hole or not lost forever, or something that is not both in a black hole and not lost. The English speaker was non-committal on the existence of things that might fall into black holes; however these last two formulas commit themselves to the existence of something. Moreover, these formulas are made true by the existence of my wallet, since it has not fallen into a black hole. The original English speaker did not intend that result. Consider the more homely example

> There is an x, such that (if x is a golden train set component, then the supplier must be able to provide replacement components x.)

Which is equivalent to saying

> There is an x (Not(x is a golden train component) or (the supplier must be able to provide replacement components x.))

In such a case, it is, of course far better to have used the universal quantifier *to convey the mandatory nature of the requirement*. When we notice that our translation is an existentially quantified conditional, "(There is some x)(If Bx then Lx)", then we should either change the conditional to a conjunction, "(There is some x)(Bx and Lx)" to suit the quantifier, or change from an existential to a universal quantifier, "(For all x)(if Bx then Lx)" to suit the operator. The first solution won't work; it commits us to the existence of things in a black hole that are lost forever; the English does not. The second one, however, is accurate, contrary to first impression. Most likely you will want to quantify over claims, in a manner which is not as strong as the proposed universal quantification or as weak as the mere existential quantification; instead you might say that - Most x(If Bx then Lx). Though the quantification range is seen to change, the structure is the same. Be aware of the structure, and don’t existentially quantify over conditional relationships, if you can make it a conjunctive relationship.

Note that existentially quantified negated conditionals are acceptable, because they are disguised conjunctions. "(There is an x) such that Not(if Bx not Lx)" is equivalent to "(There is an x) such that (Bx and ~Lx)", which says that there is something in a black hole that is not lost forever. This is not what the original example said, of course, but it would be a good translation of the negation of the English original.

By the same token, existentially quantified negated conjunctions are usually inaccurate translations, since they are equivalent to conditionals (one conjunct as the antecedent, the other conjunct, negated, as the consequent). Don't mistake existentially quantified conditionals (which you should generally avoid) for existentially quantified antecedents to conditionals (which are perfectly acceptable). An example of the latter is "If there is a God, then I'm in trouble": "(There is some x)(If Gx ,then Ti)". The scope of the existential quantifier only extends to the antecedent, not the entire conditional. Expressions like this help us translate sentences that say, in effect "we are non-committal on the existence of x, but if we do commit ourselves to the existence of x, then..." . This will be especially useful in the writing of optional requirements.

If there is a golden toilet, then the supplier should install the golden toilet in the driver's cabin.

It follows that there are two ways to be non-committal about the existence of something: (1) use a universal quantifier, or (2) use an existential quantifier in the antecedent of a conditional.

Rules for the semantics

- Always track the interactions between the effect of the operations on each other
- Use the definitions of the connectives as your guide. Pay attention to the tips
- Be careful with Brackets, be sure to identify the scope of any operation (e.g. negation, quantification, etc) in any sentence you are reading or writing.
- Avoid negated sentences, if possible try to restate them positively.
- Seek to reduce the number of explicit quantifiers in any way you can.
- By far, the preferred structures for writing requirements are universal conditionals and existential conjunctions. The latter allow us to state particulars, and the former allow us to state principles.
- Think about the sentence structure of each sentence you plan on writing. Can we easily evaluate with these rules? If not, either you haven't thought about it properly or your sentence is nonsensical. In either case think harder.

7 Combining syntax and semantics

In this chapter we combine the theory of chapter (4), (5) and (6) in a step by step procedure for determining when a requirement has been fulfilled. This should be seen as an *ideal* procedure, the theory of the previous chapters is applied consequently in an ideal way. However, in practise you won't have the time to go through all steps of the procedure for each requirement. Therefore in chapter (8) we will explain up to what point the procedure is useful for making better requirements *in practice*.

However, the main point of this chapter is to elaborate a step by step procedure for the testing of requirement satisfaction. We will show, with an example, how to assess complex requirements for compliance.

We will now focus on one example to explain the ideal procedure according to our theory. In order to assess whether a sentence is constructed in a correct, logical order, it helps to dissect the complex sentence in the simple sentences that together make up that complex sentence. This can be done by using brackets. Consider

1. Nixon won the election and lied to the people, he covered it up, and then died.

…a sentence with a clear ambiguity over what we mean by "it". Now examine (2).

2. Nixon won the election and lied to the people, he covered up the lie, and then died

…in this sentence we have added a reasonable assumption about what "it" referred to in (1).

3. ((((Nixon won the election), and There are some x (x are the people, and Nixon lied to x)) and (There is something y (y is a lie and Nixon covered up y))) and then (Nixon died))

…while in this sentence we have simply identified the sentence forms of the atomic and complex sentences included and bracketed them.

4. ((Nixon won the election) and There are some x (Nixon lied to x) and There is something y (y is a lie and Nixon covered up y)). (Nixon died.)

…we have now noticed that the idiomatic "and then" cannot serve as a connective in our logical language, because we have no way to express temporal ordering.

5. (((WON(n) and There are x (P(x) and (LIED(n)(x))) and There is an y (LIE(y) and CU(n)(y))), (Dn).

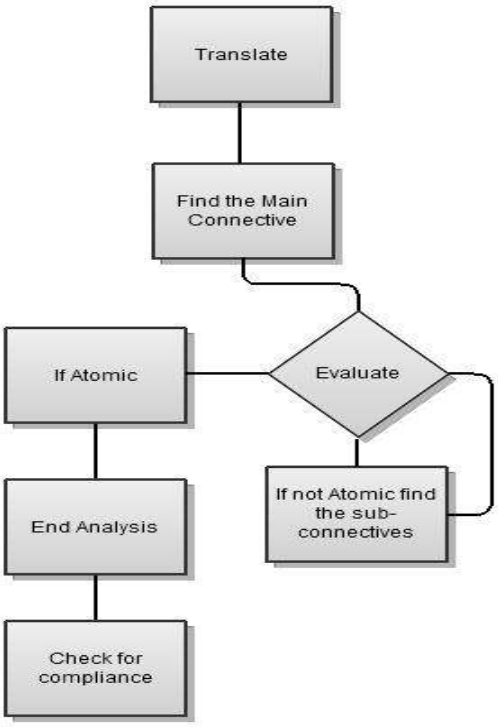
Note how stepping from (1) to (2) requires that we make some assumptions about relevance. *Can you see why, if so, can you also explain why quantifiers are important?* This is a common practice. More importantly, notice how the final form shows that the first sentence was in fact two ambiguous sentences. Crucially perhaps, you should note that the presence of the word "then" does not effect the logical form of the sentence because we did not say that Nixon's death was conditional on his lying, we merely said he died. Note, the sentence strongly implies a temporal order between election and death, but our logical language is too poor to express this properly. It's possible he could have died before being elected. Also if we wanted to say that Nixon died from his lying, we would need to change the original sentence.

1. Nixon won the election and lied to the people, and if he covered up the lie then he will die.
2. (Nixon won the election) and There are some x (Nixon lied to x) and If (There is something y (y is a lie and Nixon covered up y)) then (Nixon will die.)
3. ((WON(n) and There are x (P(x) and (LIED(n)(x))) and (if (There is an y (LIE(y) and CU(n)(y))) then (Dn)).

The examples above show that a complex sentence can be divided in two or more simple sentences and that these sentences are connected to each other using connectives. The depth of analysis you

go to should be such that it helps clear up the structure of the sentence. More generally, we can now see that dependant on the meaning of the main connectives, the sentence acquires a certain meaning and the meaning can change when other connectives are used. The meaning of the connectives is defined in such a way so that we can show when and where the connective is appropriately used. There is a step by step procedure for determining when a sentence has been fulfilled. This is exact same process you will use to evaluate requirements.

First we depict the process graphically, and then explain each step.



For purposes of illustration we will stick with the Nixon example. Step (1) is already done, as you can see above.

So we have

Step 1: Translate

((WON(n) and There are x (P(x) and (LIED(n)(x))) and (if (There is an y(LIE(y) and CU(n)(y))) then (Dn)).

Step 2: Identify the main connective.

Since the main connective is the one with the widest scope, we know that:

((WON(n) and There are x (P(x) and (LIED(n)(x))) **and** (if (There is an y(LIE(x) and CU(n)(y))) then (Dn))

Step 3: Evaluate:

By the definition of conjunction, it follows that

((WON(n) and There are x (P(x) and (LIED(n)(x))) **and** (if (There is an y(LIE(y) and CU(n)(y))) then (Dn))

Is true if and only if

(a) WON(n) and There are x (P(x) and (LIED)(n)(x)) is true

and

(b) If(There is an y(LIE(y) and CU(n)(y))) then D(n) is true

Now we need to determine if (a) and (b) are true. To do so, we first need to identify the main connective of (a) and (b)

Step 4: Identify the sub-connectives

For (a) the main connective is the conjunction **_and_,** but for (b) the main connective is the **if_then_** connective.

Step 5: Evaluate

By the conjunctive definition

(a) WON(n) and There are x (P(x) and (LIED)(n)(x)) is true

If and only if

(s) WON(n) is true

(t) There are x(Px and (Lied(n)(x)) is true

And

By the conditional definition

(b) If(There is an y(LIE(y) and CU(n)(y))) then D(n) is true

If and only if either

(u) There is an y(LIE(y) and CU(n)(y)) is false

OR

(v) There is an y(LIE(y) and Cu(n)(y)) is true and D(n) is true.

At this stage, we still need to evaluate (s), (t) (u) and (v), but the procedure is the same: find the main connectives of each and apply the definition. You should find that the entire sentence will be true if and only if (s) Nixon won the election and (t) there are some people that Nixon lied to, and (u) if there was some lie that Nixon covered up, he (v) died. Some very small knowledge of history and science allows you to assess this sentence as true, because it follows from the fact that every man dies, that Nixon will die, so the conditional (b) is true whether (v) is true or not. Also, we know that (a) is true simply by looking up any standard history of the time. So since (a) and (b) are true, the entire sentence

((WON(n) and There are x(P(x) and (LIED(n)(x))) **and** (if (There is an y(LIE(y) and CU(n)(y))) then (Dn))

is true.

Testing atomic sentences: Checking For Compliance

Every complex sentence is easily evaluated by applying the definitions, but when it comes to testing whether the atomic sentences are true, we need to check our sentences against the universe of discourse, the things we are talking about. For most practical purposes for NS, this simply means checking to see if the supplier has implemented the stated requirements in their design. In chapter 7 we will explicate on verification of requirements, and making verification requirements. However the idea to keep in mind at the moment is that once the atomic sentences have been checked for compliance we can build up the sentence with which we started filling in the truth values for the increasingly complex sentences to determine whether the entire sentence is true.

Meeting Requirements: Implementing design

If we wanted to invert this procedure with an aim to comply with some requirement rather then check for compliance. We need only take some statement of a requirement and break it down to atomics, determine whether the rules are such that we (as suppliers) need to comply with every atomic, or whether the structure of requirement is such that some atomic components are optional.

In this way both NS and the supplier can be assured that just so long as NS writes its requirements in a way which logically ensures that its objectives need (or might) be met, then following these simple rules will result in a convergence between the desires of NS and the practice of the supplier. For this reason it is absolutely crucial to understand the role of the conditional and the disjunction. The former allows you express necessary and sufficient claims, while the latter allows you to clearly present options.

Key point

We have presented the ideal way to determining when a requirement has been fulfilled. However, applying this method on all requirements would take too much time, and therefore we will explicate on a practical method in the following chapter. The key point of our step by step method is **step 2: identify the main connective**. Find the main connective, and you find the failure conditions. Therefore we will focus in finding the main connective next chapter.

Rules for combining syntax and semantics

- In order to determine when a requirement has been fulfilled, apply the step by step procedure!
- Translation of natural language into a logical form is a skill that develops with practice, so practice. If you think logically, you can write logically.

8 Identifying failure – finding the main operation

In this chapter we will return to some examples from chapter 2, and we will show you how to apply the key step from the procedure: finding the main operation of a requirement. We include, both the original format and the proposed rewrite, as you can see the rewrite aims to take advantage of the semantics of our system, both *n*i and *n*ii are equivalent but only the latter is easily evaluated.

Examples

1i The passenger emergency brake system shall, in the case of a passenger's emergency brake application, indicate in the cab which emergency brake handle has been operated.

1ii If the passenger emergency brake has been triggered, it shall be shown in the cab, that (a) an emergency brake has been triggered and (b) which emergency brake was triggered.

Semi-Formally,

If(the passenger emergency brake has been triggered) then (it shall be shown in the cab that the emergency brake has been triggered) and (it shall be shown in the cab which emergency brake was triggered).

Now, what is the failure condition of the claim? Simple, we just refer to the semantic rules for the conditional. In other words, the requirement has been fulfilled just where either (i) the passenger emergency brake is never triggered, or (ii) whenever the passenger emergency has been triggered, both the fact that a passenger emergency brake has been triggered, and which brake was triggered are shown in the cab.

3i In case of a turned off heating system, the toilet shall withstand an external temperature of -10°C for a continuous period of 12 hours without damage , assuming that the internal temperature of the vehicles was approximately +20°C before the heating system was turned off.

3ii If the heating system is turned off and the internal temperature of the vehicles was approximately +20°C before the heating system was turned off, the toilet shall withstand an external temperature of -10°C for a continuous period of 12 hours without damage.

Semi-Formally,

If (the heating system is turned off) and (the internal temperature of the vehicles was approximately +20°C before the heating system was turned off) then (the toilet shall withstand an external temperature of -10°C for a continuous period of 12 hours without damage).

For this example, we main operation is also the conditional. Again, when investigating the failure condition of the claim, we have to refer to the semantic rules for the conditional. This requirement has been fulfilled just where either (i) it is not the case that (the heating system is turned off) and (the internal temperature of the vehicles was approximately +20°C before the heating system was turned off) or (ii) (the heating system is turned off) and (the internal temperature of the vehicles was approximately +20°C before the heating system was turned off) and (the toilet withstands an external temperature of -10°C for a continuous period of 12 hours without damage).

4i Dedicated personnel shall be able to set the HVAC system to a “cleaning” or “anti-freeze” mode during which the heating/ventilation system regulates to an interior temperature of 5°C and the cooling system is blocked.

4ii Dedicated personnel shall be able to set the HVAC system to a "cleaning" or "anti-freeze". In either mode, the cooling system shall be blocked, and the heating/ventilation system shall regulate the interior temperature to 5°C exactly.

We should note this is actually two requirements.

The first semi- formally,

For all x, such that x is one of the dedicated personnel and there is a y such that y is a HVAC system, then x shall be able to be set y into "cleaning " or anti-freeze mode.

This shows that the sentence is a universal claim of the form

For all x, (if((x is D) and there is a y (y is H)), then there is a mode C, and there is a mode A, such that (If (Sxy), then possibly (Cy or Ay))

In this example, we have quantified over the objects x, and y, and the properties of being in a mode C, or A. This is perfectly okay, and we do it all the time, but it can be tricky to write clearly about which properties you've quantified over, so be careful. However, it remains easy to see the failure conditions of this sentence. This is a universal claim about the capacities available to any dedicated staff member, as such if at any point (even) one dedicated staff cannot interact with the HVAC system as specified, the requirement has been failed.

We have a choice when writing the second sentence. Either we can, for clarity's sake restate the object of our discussion, or rely on the fact that we can refer backwards to the object The latter options means we have to use an indefinite referring term, The options are as follows

Semi-formally

4iii There is an x, such that x is a HVAC system, If (x is in mode C, or x is in mode A) then, (the cooling system shall be blocked) and (the heating/ventilation system shall regulate the interior temperature to 5°C exactly).

Here it is obvious that the main operation is a conditional, since its scope is by far the largest. This is appropriate. Consider the other option where we rely on the earlier sentence, and refer backwards to the indefinite object "y" discussed in 5i and 5ii).

4iv If (y is in mode C, or y is in mode A) then, (the cooling system shall be blocked) and (the heating/ventilation system shall regulate the interior temperature to 5°C exactly).

As such, the main operation is the same conditional format, and the failure conditions are obvious.

A moral about indefinite reference

The re-specification of the object in (4i) adds length but also clarity. Unless you are very sure that referring backwards, by means of phrases, such as "that x", "those y" "it"…etc, causes no risk of ambiguity you should always introduce the object of discussion. Naturally we assume that if there is any object introduced to the discussion

There is an x such that Fx….

…then the indefinite object of all subsequent sentences is thought to be identifiable with x, unless otherwise specified, or a new object is introduced.

There is an x such that F…..

……

There is a ***(new!)*** x such that Gx.

Objects can be introduced haphazardly, so it is crucial that we pay attention to when the object of discussion has changed. If it has changed, are we sure the change has been adequately highlighted?

Example(s)

Consider the dialogue.

> A: You should go to the Leonard Cohen concert tomorrow.
>
> B: I can afford it, but I can't justify it.

The claim made by B is ambiguous in at least three ways, because of the ambiguity of "it".

1st Reading: There is a concert, I can afford to go, but I can't justify going.

2nd Reading: There is a concert, I can afford the ticket but I can't justify buying it.

3rd Reading: There is a concert, I can afford it, but I can't justify why I can afford it.

Consider another claim

> The current NS rolling stock is fitted with locks consisting of the CM99 locking system of the company DOM, Rijswijk, the Netherlands. SNG shall follow the same principles.

This claim assumes that anyone reading it will know the nature and governing principles of the CM99 locking system. However, this is not an assumption that can be sustained in practice. So despite including a name, the sentence is ambiguous. Always be sure, whenever you introduce an object, you directly characterize the object or indicate where a detailed characterization can be found. In addition, wherever you refer to a previously characterized object, be clear about which object that is. Otherwise it will be difficult to determine exact failure conditions since an ambiguous sentence can be said to fail in many ways.

9 Structuring a requirement specification logically

It is important to choose one standard way of clustering the requirements and the related sentences. In this chapter we will explain the preferred method to order a requirements specification. This method allows us to order our concerns in such a way that the whole document presents the requirements and their relations in an unambiguous way.

9.1 Definitions and abbreviations

Requirements should *always* be preceded by the definitions of all words and terms for which reliance on dictionary definitions is not appropriate. Preferably these definitions are given in a separate section "Definitions and abbreviations", just before the section with requirements. In particular the following categories of terms should be defined:

1. Terms which are not likely to be in the vocabulary of the intended users of the specification.
2. Terms which have multiple dictionary meanings but only a single specification meaning.
3. Technical terms.
4. Terms which are used with special meanings.

For example, the definitions of "shall", "should", "may" and "will" should always be given in this section, since they have multiple dictionary meanings but only a single specification meaning (category 2) and they are used with special meanings in requirement making (category 4). For instance, the following definitions for "shall" and "may" can be given:

> "Shall" expresses a characteristic which is to be present in the item which is the subject of the specification, i.e. "shall" expresses a binding requirement.
>
> "May" expresses permissive guidance.

For completeness, in this section NS should refer to the dictionary to be used for all the terms which are not defined. Furthermore, all abbreviations and acronyms used in the requirements should be listed alphabetically.

9.2 Specification body

The section "Specification body" should specify and order the *system requirements*, those characteristics of the system that are required to be present in the final product, together with the important information belonging to these requirements. It is preferred to use a spreadsheet for making the specification body. Regarding the horizontal structure, there should be one column for the requirements, and columns with objectives and verification requirements. Depending on the specification, more columns can be added (e.g. verification methods and standards). Each row should be numbered and classified according to the type of requirement using two different classifications. Regarding the vertical structure, requirements of the system as a whole must be distinguished from requirements of subordinate systems, and the latter requirements should be ordered with a product-structure. Below we will explicate on all aspects of the specification body just mentioned. But first we will show a preview of the preferred structure:

Required/ Optional/ Design Rec.	Number	Product (type)	Satisfaction (type)	Functionality (type)	Requirement	Objective
	4.04				**Lighting**	
RE	4.04.01	Lighting	Safety & health	Performance	**The lighting of the toilet shall be according to EN 13272.**	**The toilet area shall be properly lit.**
RE	4.04.02	Lighting	Safety & health	Performance	**The emergency lighting of the toilet shall be according to EN 13272.**	**The toilet area shall be properly lit.**
	4.05				**Toilet paper dispenser**	
OP	4.05.01	Toilet paper dispenser	Maintainability	Design	**The toilet paper dispenser shall fit rolls of toilet paper according to UIC Code 563.**	**Standard toilet paper rolls shall fit in the toilet paper dispenser.**
RE	4.05.02	Toilet paper dispenser	Availability	Design	**If standard rolls of toilet paper are used, the toilet paper dispenser shall have space for at least two rolls of toilet paper.**	**There shall be a sufficient amount of toilet paper available for passengers.**
RE	4.05.03	Toilet paper dispenser	Availability	Functional	**If standard rolls of toilet paper are used, the second roll of toilet paper shall only become available to passengers when the first roll is finished.**	**There shall be a sufficient amount of toilet paper available for passengers. Misuse of toilet paper shall be prevented.**
	4.06				**Wash bowl unit**	
RE	4.06.01	Wash bowl unit	Passenger satisfaction	Design	**The wash bowl unit shall be provided with a drain with a net.**	**The wash bowl outlet shall have a high impact against cloaking. The grey water system shall not cause any unpleasant odours inside the toilet. e.g siphon.**
RE	4.06.02	Wash bowl unit	Passenger satisfaction	Functional	**The wash bowl unit shall be provided with components to prevent bad odours entering the toilet trough the grey water system.**	**It shall not be possible for bad odours from the waste water system to enter the toilet trough the wash bowl unit.**
RE	4.06.03	Wash bowl unit	Passenger satisfaction	Functional	**The wash bowl unit shall be provided with components to prevent outside noise entering the toilet through the grey water system.**	**It shall not be possible for outside noise to enter the toilet trough the wash bowl unit.**
RE	4.06.04	Wash bowl unit	Passenger satisfaction	Functional	**The wash bowl unit shall be provided with components to prevent bacteria contamination.**	**Bacteria from the waste water system shall not contaminate the wash bowl unit and the toilet.**
RE	4.06.05	Wash bowl unit	Passenger satisfaction	Functional	**The wash bowl unit shall be provided without a plug.**	**A bowl-overflow shall be prevented.**
RE	4.06.06	Wash bowl unit	Sustainability	Performance	**The washbowl unit and its fixings shall withstand a static force of 1000 N without damage. The load may be applied at any accessible part of the washbowl unit. An application surface of approximately 50x50 or Ø 55 mm shall be used.**	**The washbowl unit shall be of sufficient strength.**
	4.07				**Soap dispenser**	
RE	4.07.01	Soap dispenser	Maintainability	Functional	**It shall be possible to access the soap dispenser with a square socket key according RIC (Berne key).**	**The soap tank shall be accessible for stocking.**
RE	4.07.02	Soap dispenser	Maintainability	Functional	**It shall be possible to refill the soap dispenser tank by pouring the soap from above from a separate container, without removing the soap dispenser tank.**	**The soap tank shall be accessible for stocking.**
RE	4.07.03	Soap dispenser	Maintainability	Design	**The soap dispenser outlet shall be positioned directly above the washbowl.**	**Pollution by dripping soap shall be prevented.**

Figure 1: Preview of the preferred structure of the specification body.

9.2.1 Objectives

Each requirement should be accompanied with an objective which expresses exactly why that particular requirement has been incorporated in the specification body. Preferably the objectives are written in the column next to the requirements. It is important that every aspect of a requirement is explicated in the objective. For example, for the following requirement from the EuroSpec toilet specification

> "When a failure or event signal is cleared, the toilet shall restart automatically. It shall be indicated for which specific failures a manual reset is required."

the following objective is adequate

> "The toilet system shall start up automatically after maintenance, resolved failure signals or resolved events. (Refilling fresh water, emptying full waste tanks, etc.) A manual reset should only be used if necessary."

since it explains all aspects of the requirement. If this is done throughout the document, the supplier will never be in doubt about the reasons for the implementation of some requirement.

Furthermore, when the supplier has an idea for improving an optional requirement, the objective can serve as a guideline. Only when all aspects of the objective are met in this idea, the supplier can discuss it with NS. We will come back to this in paragraph 9.2.3.

9.2.2 Verification criteria

Each requirement should be accompanied with "verification requirements", written in a separate column. In a verification requirement, NS should define the quality of evidence that the system has to met. The theory presented in chapters 5 and 6 is important in determining the appropriate

verification requirements, since we should know the failure conditions of some requirement in order to prescribe how (and up to what level of evidence) it should be verified by the supplier.

NS may also state for each requirement the method(s) to be used to prove that the requirement has been met, which are called "verification methods". If used, these verification methods should be written in the column next to the verification requirements. Examples of verification methods are:

1. **Test.** The operation of the system, or a part of the system, using instrumentation or other special test equipment to collect data for later evaluation;
2. **Demonstration.** The operation of the system, or a part of the system, that relies on observable functional operation not requiring the use of instrumentation, special test equipment, or subsequent analysis;
3. **Analysis.** The processing of accumulated data obtained from other qualification methods. Examples are reduction, interpolation, or extrapolation of test results;
4. **Inspection.** The visual examination of system components, documentation, etc.;
5. **Analogy.** The use of evidence from verification of an analogous product, for example a prior version;
6. **Certification.** A declaration by a designated stakeholder, usually the supplier or developer;
7. **Special verification methods.** Any special qualification methods for the system, such as special tools, techniques, procedures, facilities, acceptance limits, use of standard samples, preproduction or periodic production samples, pilot models, or pilot lots.

Where sampling is to be used, the rules for selecting samples should be given.

9.2.3 Degree of detail

Regarding the degree of detail to be incorporated in specifying requirements, those characteristics of the system that are necessary for the system to satisfy its intended use should be included in an unambiguous form. In determining these characteristics the criterion which should be used is the level of risk associated with the ideal "that any system which is supplied which satisfies the requirements, will satisfy the need". The level of acceptable risk with respect to attainment of this ideal should be determined as a prerequisite to preparation of the system specification. Those characteristics that NS is willing to leave up for the supplier, should either be omitted or be specified in the form of an option. From the introduction to the supplier it should be clear that optional requirements can be ignored, if the supplier has a better idea to fulfil the corresponding objective.

An example of an optional requirement in the EuroSpec toilet specification is:

> "The source code and compilation tool of the controller shall be provided."

However, it is well-known that many suppliers don't want to provide the NS with their source codes. Such a supplier should read the objective carefully:

> "Any function modifications of the controller shall be possible, during the whole lifetime of the train. New development of toilet controllers or future upgrade shall be possible. Demonstration of the level of security of the system is available."

The supplier may find a solution for which this objective will be met, without having to provide the source code. For example, the supplier can store all the knowledge carefully and guarantee that whenever NS wants to make function modifications or wants to upgrade the controller, this can be done by the supplier.

9.2.4 Requirements of the system as a whole and of subordinate systems

There are several ways to cluster the system requirements logically. For complex systems, it is important to *distinguish between requirements of the system as a whole and requirements of a*

subordinate system. The requirements of the system as a whole should be presented first, in the subsection "Requirements of the system as a whole". They should be labelled according to "requirement type"; we will explicate on this in paragraph 9.2.5.

The requirements of subordinate systems should be presented in different subsections, one subsection for each subsystem, component or product for which there are corresponding requirements. The structure arising from this method is called a "product structure". For example, the EuroSpec toilet specification document can be divided into the following subsections: "Toilet module", "Lightning", "Toilet paper dispenser", "Wash bowl unit", "Soap dispenser", "Hand dryer", "Waste bin", "Toilet bowl", "Fresh water system", "Waste water system", "Toilet system", "Diagnostic system", "Controller", "Doors and locks". The requirements of subordinate systems should also be labelled according to requirement type, as we will explain in paragraph 9.2.5.

9.2.5 Structuring according to type of requirement

It is important to distinguish between types of requirements, because requirements which are separated by the product-structure should be logically integrated on a higher level. By labelling requirements according to type distinctions, different parties which are involved in the process of requirement construction and interpretation can see for each requirement what kind of requirement they are dealing with. Moreover, they can order the spreadsheet according to the type distinction of their interest, to see the requirements specification on the desired level. Before providing you with examples, we will explain the two type distinctions which should be used.

The goal, as we see it, is to write requirements with a mind to (a) the direct in-use purpose of the objects/system described, and (b) the expectations of the NS. We think it would be profitable to index each requirement with two requirement types, the first should specify the kind of requirement it is, i.e. the end functionality (type 1). The second should specify the kind of expectation NS deems the requirement to satisfy i.e. its "-ility" (type 2).

Type distinction 1: functionality

- An **external interface requirement** states the required characteristics of an item/system at a localised point, or region, of connection of the item/system to the outside world. An example from the EuroSpec toilet specification is:

  "The positioning of the interface for servicing the waste tanks shall be according to Appendix 9 of UIC Code 563".[10]

- A **functional requirement** states what the item is to do. An example from the EuroSpec toilet specification is:

  "The waste bin shall withstand typical waste originating from the use of the toilet."[11]

- A **performance requirement**, for a given function, states how well that function is to be performed by the item. An example from the EuroSpec toilet specification is:

  "The manual draining flow of the fresh water tank shall be minimum 0,7l/s."[12]

- An **environmental requirement** limits the effect that the external enveloping environment (natural or induced) is to have on the item, and the effect that the item is to have on the external enveloping environment. An example from the EuroSpec toilet specification is:

  "If the heating system is turned off and the internal temperature of vehicles is approximately +20°C when the heating system is turned off, the toilet shall withstand an external temperature of -10°C for a continuous period of 12 hours without damage."[13]

[10] Source: EuroSpec *Specification for Toilets of Railway Vehicles,* 1st edition, April 2012.
[11] Source: EuroSpec *Specification for Toilets of Railway Vehicles,* 1st edition, April 2012.
[12] Source: EuroSpec *Specification for Toilets of Railway Vehicles,* 1st edition, April 2012.

- A **resource requirement** limits the usage or consumption by the item of an externally provided resource. An example from the SNG wheel sets specification is:

  "The Train Sets shall be prepared for conversion to mono-current Train Sets for a traction power supply of 3000 V DC."[14]

- A **physical requirement** states the required physical characteristics (properties of matter) of the item as a whole (e.g. mass, dimension, volume.) An example from the SNG high voltage installation specification is:

  "In case of the use of solid wheels, the wheels shall comply with EN 13262 and be made out of material ER7 or ER8 according to EN13262, or of a material with equivalent mechanical properties."[15]

- A **design requirement** is used to specify aspects that should be either (a) incorporated into the design of the product or (b) explicitly not incorporated in the design of the product. An example from the EuroSpec toilet specification is:

  "Components that are difficult to clean shall not be located directly under the waste water level sensors."[16]

NS should use a separate column, preceding the column with requirements, where the type of each requirement is written according to this distinction.

Type distinction 2: satisfaction

- **Availability**. An example from the EuroSpec toilet specification is:

  "It shall be possible to flush the toilet system every 90 seconds. A second flush shall be possible within 30 seconds after the first flush."[17]

- **Reliability**. An example from the EuroSpec toilet specification is:

  "The inner diameter of the piping of the waste water system shall not shrink in width, from the toilet bowl to the waste tank."[18]

- **Maintainability**. An example from the EuroSpec toilet specification is:

  "It shall be possible to adjust parameters related to the operation of the toilet. This shall be possible using a standard laptop that has been installed with the required software."[19]

- **Passenger satisfaction**. An example from the EuroSpec toilet specification is:

  "If hinged toilet doors are used, they shall rotate into the toilet module when opened."[20]

- **Sustainability**. An example from the EuroSpec toilet specification is:

  "Filling fresh water tanks shall not damage the water tanks."[21]

- **Safety & health**. An example from the EuroSpec toilet specification is:

  "The train staff shall be able to view the entire toilet area while standing in the door opening."[22]

[13] Source: EuroSpec *Specification for Toilets of Railway Vehicles*, 1st edition, April 2012.
[14] Source: section 6 SNG early draft.
[15] Source: section 6 SNG early draft.
[16] Source: EuroSpec *Specification for Toilets of Railway Vehicles*, 1st edition, April 2012.
[17] Source: EuroSpec *Specification for Toilets of Railway Vehicles*, 1st edition, April 2012.
[18] Source: EuroSpec *Specification for Toilets of Railway Vehicles*, 1st edition, April 2012.
[19] Source: EuroSpec *Specification for Toilets of Railway Vehicles*, 1st edition, April 2012.
[20] Source: EuroSpec *Specification for Toilets of Railway Vehicles*, 1st edition, April 2012.
[21] Source: EuroSpec *Specification for Toilets of Railway Vehicles*, 1st edition, April 2012.

More categories of satisfaction, mostly "–ilities", can be added to this type distinction, e.g. reusability and transportability. NS should use a separate column, preceding the column with requirements, where the type of each requirement is written according to this distinction.

The purpose of indexing the requirements in this manner is to encourage conformity between the expectations of NS at the level of product design, and broader goals of product performance and upkeep. If we treat these type-distinctions as two axes which are designed to converge at an end point, we can expect the supplier to treat the two complimentary aspects of design as equally important. Furthermore it is possible for the tenderers, and finally for the supplier, to order the list of requirements according to the type distinction (or column) of interest.

As an example, consider the following requirement:

> "The manual draining flow of the fresh water tank shall be minimum 0,7l/s."

This requirement belongs to a subordinate system, namely the fresh water system, and therefore it belongs to the subsection "Fresh water system". With regard to type distinction 1, it is a performance requirement, and with regard to type distinction 2, it is a maintainability requirement.

Principles

Ultimately, the objective of requirement writing is to make clear the kind of product we want, but also what we expect of the product. The requirements which specify particular functionalities of the product only do half the job, we also need to convey when our expectations have been met.
The two type distinctions may also be used to establish a metric for determining what counts as a principle endorsed by NS i.e. a general statement that should govern the design and implementation of a requested product. For any set of requirement specifications relating to one system or object, we can observe the number of functionality or satisfaction requirements and abstract from this, the principle that (a) some aspect of functionality or (b) some aspect of NS satisfaction is of paramount importance. We would use various considerations, for example, of the numeric superiority of functionality requirements over satisfaction requirements, or the conjunction of environmental requirements with sustainability requirements, etc. The utility of indexing each requirement is that we can then observe what are the broadly prevalent concerns of the requirement authors and come to agree sound principles of design and implementation for endorsement.

Rules regarding the structure of the specification body

- Incorporate a list of definitions and abbreviations in the requirements specification, preceding the specification body.
- Make sure that each requirement is accompanied with an objective which expresses *exactly* why that particular requirement has been incorporated in the specification body.
- Incorporate verification requirements, and if possible verification methods, in the specification body.
- Distinguish between requirements of the system as a whole and requirements of a subordinate system.
- Distinguish between types of requirements according to the functionality- and the satisfaction-distinction.

[22] Source: EuroSpec *Specification for Toilets of Railway Vehicles,* 1st edition, April 2012.

10 Appendix

This section introduces a way to graphical define the logical connective so as to compliment the rules elaborated in chapter five. There will be an illustrative picture for each rule except for the quantifier rules.

P	Not(P)
1	0
0	1

P	Q	(P or Q)
1	1	1
1	0	1
0	1	1
0	0	0

P	Q	(P and Q)
1	1	1
1	0	0
0	1	0
0	0	0

P	Q	(if P then Q)
1	1	1
1	0	0
0	1	1
0	0	1

The quantifiers rules are to be understood only with respect to a particular universe of discourse, as such the rules given in chapter (5) are schematic and remain so. Please consult chapter (5) when using quantifiers.

Colofon

Author(s)	Nathaniel Forde, Aafke de Vos
Reference	
Date	August 24, 2012
Version	1.0
Status	Final
File	\\prod.ns.nl\USR\user12\Aafke.deVos\Documenten\Manual Writing Requirements 24-08-2012.doc