Application of Requirements Analysis Techniques to the analysis of civil aviation security standards

Régine Laleau[a], Sylvie Vignes[b], Yves Ledru[c], Michel Lemoine[d], Didier Bert[c], Véronique Donzeau-Gouge[e], Catherine Dubois[e] and Fabien Peureux[f]

[a] LACL, Université Paris12, Paris, France

[b] GET/ENST, Département Informatique et Réseaux, Paris, France

[c] LSR/IMAG, Grenoble, France

[d] ONERA/CdT/SAE, Toulouse, France

[e] CEDRIC/CNAM, Paris, France

[f] LIFC, Besançon, France

Abstract

The security of civil aviation, like many human activities, is regulated by a series of international standards and recommended practices. The quality of these documents is a prerequisite to effectively reach an acceptable security level. The EDEMOI[1] project aims to integrate and apply several requirements engineering and formal methods techniques to analyze these standards. The project applies a two-step approach. In a first step, properties are extracted from the natural language standard and integrated in a domain model; a conceptual model of the underlying system is also elaborated. In a second step, the properties and the models are transformed into a formal model which will be analyzed by formal methods tools. The paper considers the standard produced by the ICAO (International Civil Aviation Organization) and focuses on the first step. We have chosen to apply existing requirements engineering techniques and the paper details the approach, presents several extensions/adaptations to the RE techniques, necessary to take into account the specific features of our application domain. Finally, we report on sample results on how formalization of some imprecise parts of the standard can improve its understanding.

Proceedings of the First International Workshop on Situational Requirements Engineering Processes: Methods, Techniques and Tools to Support Situation-Specific Requirements Engineering Processes (SREP'05), Organized by IFIP WG8.1 Method Engineering Task Group, In conjunction with the 13th IEEE International Requirements Engineering Conference. Paris, France, 29-30 August 2005. pp. 91-106

[1] The EDEMOI project is supported by the French National Action Concertée Incitative "Sécurité Informatique".

Application of requirements engineering techniques to the analysis of civil aviation security standards

Régine Laleau[a], Sylvie Vignes[b], Yves Ledru[c], Michel Lemoine[d], Didier Bert[c], Véronique Donzeau-Gouge[e], Catherine Dubois[e] and Fabien Peureux[f]

[a] LACL, Université Paris12, Paris, France

[b] GET/ENST, Département Informatique et Réseaux, Paris, France

[c] LSR/IMAG, Grenoble, France

[d] ONERA/CdT/SAE, Toulouse, France

[e] CEDRIC/CNAM, Paris, France

[f] LIFC, Besançon, France

Abstract

The security of civil aviation, like many human activities, is regulated by a series of international standards and recommended practices. The quality of these documents is a prerequisite to effectively reach an acceptable security level. The EDEMOI[1] project aims to integrate and apply several requirements engineering and formal methods techniques to analyze these standards. The project applies a two-step approach. In a first step, properties are extracted from the natural language standard and integrated in a domain model; a conceptual model of the underlying system is also elaborated. In a second step, the properties and the models are transformed into a formal model which will be analyzed by formal methods tools. The paper considers the standard produced by the ICAO (International Civil Aviation Organization) and focuses on the first step. We have chosen to apply existing requirements engineering techniques and the paper details the approach, presents several extensions/adaptations to the RE techniques, necessary to take into account the specific features of our application domain. Finally, we report on sample results on how formalization of some imprecise parts of the standard can improve its understanding.

Keywords: civil aviation standard modelling, security goals, UML notation, formal notation

1 Introduction

The security of civil aviation is governed by a series of international standards and recommended practices that detail the responsibilities of the various stake-holders (states, operators, agents, ...). These documents give the specifications of various procedures and artifacts which implement security in airports, aircrafts and air traffic control.

A key element to enforce security is the conformance of these procedures and artifacts to the specifications. However, it is also essential to ensure the consistency and completeness of the specifications. Standards and recommended practices are natural language documents (written in English or French) and their size may range from a few dozen to several hundred pages. Natural language has the advantage to be easily understood by a large number of stake-holders, but practice has also shown that it can be interpreted in several inconsistent ways by various readers. Moreover, it is very

[1] The EDEMOI project is supported by the French National Action Concertée Incitative "Sécurité Informatique".

difficult to process automatically natural language documents in the search for inconsistencies. When a document has several hundred pages, it is very difficult to ensure that the content of some paragraph is not contradicted by some other paragraphs distant by several dozens of pages from the first one.

The EDEMOI project (Edemoi, 2004) aims to provide an industrial methodology and tools to build models from these natural language documents, and use specialized tools to study their consistency. These models will also be used to build test cases to assess the conformance of a given implementation to the security standards. The project applies a two-step approach. In a first step, standards described in natural language are analyzed in order to extract security properties and to elaborate a conceptual model of the underlying system. The second step consists in analyzing and verifying the model by different kinds of formal methods.

The paper considers a standard produced by the ICAO (International Civil Aviation Organization) and focuses on the first step. It describes and explains how we have applied existing requirements engineering (RE) methods, and specially goal-oriented methods, to elaborate the model of the standard and presents a number of extensions/adaptations we have brought to them in order to take into account the characteristics of the domain. The use of these methods has allowed us to highlight imprecisions and/or deficiencies in the international standard and we show how the formalization of some imprecise parts of the document can improve its understanding.

The paper is organized as follows. Section 2 presents the context of airport security and the different standards. Section 3 summarizes the objectives of the EDEMOI project. Section 4 presents the approach to analysis standards and the adaptation of RE techniques. Section 5 reports on sample results of problems we have encountered and how we can take profit of formal notations. The paper concludes with some remarks about the results and future directions.

2 Context: Airport Security

The airport is actually the place where passengers and their baggages are controlled before boarding an aircraft. Preventing dangerous objects from being brought on board an aircraft is a significant step to prevent acts of unlawful interference during a flight.

Airport security controls are governed by local regulations that inherit from national and international standards.

1. At the international level, the International Civil Aviation Organization (ICAO) has produced an international standard and recommended practices related to security, called Annex 17 (A17, 2002). It is a very general document that must be followed by all countries members of the ICAO.

2. At the European level (45 countries), the European Civil Aviation Conference (ECAC) is in charge of refining these rules into a more detailed standard and to organize the inspection of airports to check their conformance to the ECAC standards.

3. At the national level, each country has to implement the international standards, taking into account national laws. In practice, it introduces a second level of refinement which makes the rules more precise and more constrained, and guides the design and processes of the airports.

4. At the airport level, the national and international regulations are put into practice, taking into account the specificities of the given airport. This is translated into "airport security programs".

All these documents are written in natural language, and it becomes difficult to assess the consistency of the whole. Moreover, since natural language leads to ambiguities, it may happen that two inspectors visiting the same airport at the same time reach contradictory conclusions on its conformance to the international regulations.

In the EDEMOI project, we have naturally begun by studying the highest level of regulations, the international standard Annex 17 (A17, 2002). Another reason is that this standard is a public document when most other regulations have a restricted access status. The primary objective of Annex 17 is to define the obligations of each State, member of the ICAO. The document is rigorously structured. It is decomposed into five chapters: definitions of terms used in the document (airside, screening, security controls, ...), general principles, organization of the member States, preventive security measures and management of response to acts of unlawful interference. Each chapter is organized as a sequence of numbered paragraphs, each of them defining an obligation. After a deep study and a consultation with the ECAC and the ICAO, we decided firstly to consider only chapters 1 and 4 which describe the actual procedures intended to prevent dangerous objects from being brought on board an aircraft, and secondly to focus on the cabin passengers, the cabin and hold baggage on their way from the check-in desk to the aircraft. We don't cover cargos, mail, movements of airplanes or other technical activities that take place in an airport.

3 The EDEMOI Approach

The EDEMOI project aims at defining an approach for the construction and analysis of a precise reference document that models and structures current standards and associated recommendations. This precise model can be exploited by the civil aviation authorities in several ways:

1. To improve the quality of the standards.
2. To provide complementary documents to the standard for training and documentation purposes.
3. To ease maintenance and evolution of standards by evaluating their impact on the essential security properties it enforces.
4. To define a rigorous inspection process for the evaluation of airport conformity to the standard. This will reduce misinterpretations by the inspectors in charge of this evaluation.
5. As a starting point for the definition of a facilitation process. The aim of this process is to accelerate passenger boarding and aircraft traffic, by reducing the number of controls while maintaining the same security level. This will decrease costs, and specially airplane tickets cost, and thus improve customer satisfaction.

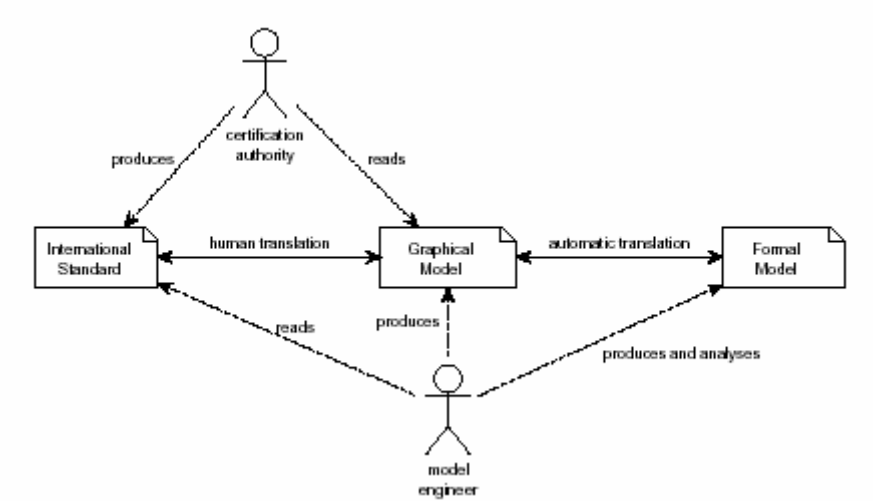

Figure 1: The Stake-holders of the modelling process

In the computer science research community, formal specification and development methods (Abrial, 1996; Spivey, 1992) have been used for system and software modelling to construct precise models and conduct detailed consistency analyzes (Lamsweerde, 2000; Ledru, 1997; Lopez et al., 2002). They provide tools to check the consistency of models, or the consistency between a model and its refinement. They can also be used as a starting point for a test generation process (Bouquet et al., 2004). We believe that such methods can ground the construction of a precise model for airport security. Moreover, their application to this project is not straightforward technology transfer, and addresses the two following research challenges.

1. Modelling airport security standards corresponds to a real size and real world problem. It will be confronted to the limits of the modelling technique, in particular regarding its expressiveness. Therefore, it is important to clearly identify the limits of the modelling techniques, and hence the limits of the formal model. Another important concern is the validation of this real size/real world problem by experts of the civil aviation authorities.

2. The model structure must distinguish between the expected security properties and the way these are implemented. Properties must be organized hierarchically and prioritized. One of the peculiarities of all the different standards and airport security programs is that they address airport security at several abstraction levels. For example, they feature general considerations (like the State obligations) but also very detailed descriptions (such as the size of passengers control equipment).

Unfortunately, formal methods and their associated mathematical language cannot be used alone. Firstly, formal methods give little guidance on constructing actual specifications. Secondly, formal models can only be read and understood by specialists. It does not really make sense to expect that readers and writers of aeronautics standards will be able to work directly with such models.

Therefore, we propose a two-step approach which takes advantage of an intermediate graphical model to support the dialog between model engineers and certification authorities. Fig.1 shows how standards are produced by certification authorities and

read by model engineers. Model engineers can then produce a graphical model of the standard, which should be read by certification authorities in order to validate the model. The graphical model can then further be translated into a formal model which will be analyzed by model engineers. Errors found in the formal document can then be propagated at the level of the graphical model to allow the discussion between model engineers and certification authorities and, if needed the correction of the standard. The first step of the approach aims at analyzing the standard and producing graphical models by using a requirements engineering process. Once formal models will be available, we will start the second step that consists in their analysis as well as the generation of test cases which can be used by inspectors to evaluate the conformance of a given airport to the international standards. The paper focuses on the first step which is detailed in the next section.

4 Regulation Analysis

As already said, Annex 17 defines the obligations of each State member of the ICAO, related to the security of international civil aviation. It does not provide to States detailed procedures to be implemented in their respective national security programs. They are the matter of other documents. Consequently, Annex 17 determines a set of preventive measures that must be applied by States in order to meet security requirements that correspond to Non-Functional Requirements (NFR). Compared with the traditional core RE activities (Nuseibeh and Easterbrook, 2000), that are eliciting, modelling and analyzing, communicating, agreeing, evolving requirements, we directly begin by the modelling and analyzing requirements step. Indeed, the eliciting requirements step is rather reduced since we have only one document that precisely describes the requirements in the form of security properties to reach.

From a methodological point of view, it seemed to us that a goal-oriented requirements process such as KAOS (Dardenne et al., 1993) or i* (Yu, 1997) was the more appropriate: identification of goals (security properties) and construction of the domain model and agent model associated to the goals. However, we need to adapt these methods to our environment. Indeed, the aim of our requirements process is very specific: it is not the elaboration of project specification that leads to the development and production of software. Our system (the airport) already exists and meets the functional requirements which it has been designed for (boarding passengers in aircrafts). What we want to do is to exhibit the security requirements described in the standards, to specify the system with formal languages, and to check that the specification satisfies these requirements by using formal proofs. Consequently the paper reports on the ability of goal-oriented RE methods, together with the necessary adaptations we have brought, to consider this unusual application domain. This is detailed in the first two subsections. The last subsection presents the model of the existing system, that is an abstract airport as it is described in Annex17.

We have used UML diagrams as the representational schema for our models (requirements and system), augmented with a number of stereotypes which allow specific features of our project to be taken into account. Although these diagrams do not convey all the details of Annex 17, they help in structuring the models and provide a support for the discussion with certification authorities and further validation of the models. There are two reasons for using UML notations. The main one is that we want to take advantage of existing tools that translate UML diagrams into formal

notations, such as B (Laleau and Mammar, 2000) or Z (Dupuy et al., 2000), or that generate test cases from UML, B or Z notations (Bouquet et al., 2004). The second reason is that we have to adopt a common notation to represent both the requirements and the system models, with regard to the certification authorities. A first specification of a concrete airport, written in UML, has already been achieved (Carton, 2004) and we will need to reuse this specification in a next step of the project when we consider refinement links between the different standards.

4.1 Security Properties (goals) Identification

The basic idea of the approach is to identify the main security properties and to analyze how these properties can be refined into sub-properties. The primary security property comes from article 2.1.1 in the second chapter of Annex 17, called "General Principles", and can be stated as follows:

> *P1: Passengers, crew, ground personnel and the general public must be safeguarded against acts of unlawful interference*

To achieve this goal, Annex 17 prescribes a set of standards and recommended practices that each State member of the ICAO has to adopt and implement. In the framework of our project we have decided to consider the preventive security measures, described in Chapter 4, the objective of which is:

> *4.1 Each Contracting State shall establish measures to prevent weapons, explosives or any dangerous device which may be used to commit an act of unlawful interference, the carriage or bearing of which is not authorized, from being introduced, by any means whatsoever, on board an aircraft engaged in international civil aviation.*

This is translated by the following security property:

> *P2: There are no unauthorized dangerous objects on board an aircraft.*

If there is an obvious causal relation between *P2* and *P1*, it needs, however, to assume the two following hypotheses:

> *H1: Acts of unlawful interference can only be committed with weapons, explosives or any other dangerous device.*

> *H2: Each State makes sure that security checks are performed in the originating state of an aircraft.*

The first hypothesis limits the kind of acts of unlawful interference by considering that only objects can be dangerous. It excludes acts coming from persons. For instance a specialist in martial art may potentially become dangerous but it is obvious that it would not be easy to detect such persons and, anyway, it is not the aim of civil aviation authorities. The second hypothesis concerns aircraft with transit stops. At a transit stop, an aircraft is considered to be secure, otherwise, it would require to control the aircraft again. However, if a state has doubts about verifications carried out by

the originating state of an aircraft, it can define specific measures. Note that whereas *H2* is clearly stated in the document, *H1* is only implicit and has been stated after discussions with the ICAO.

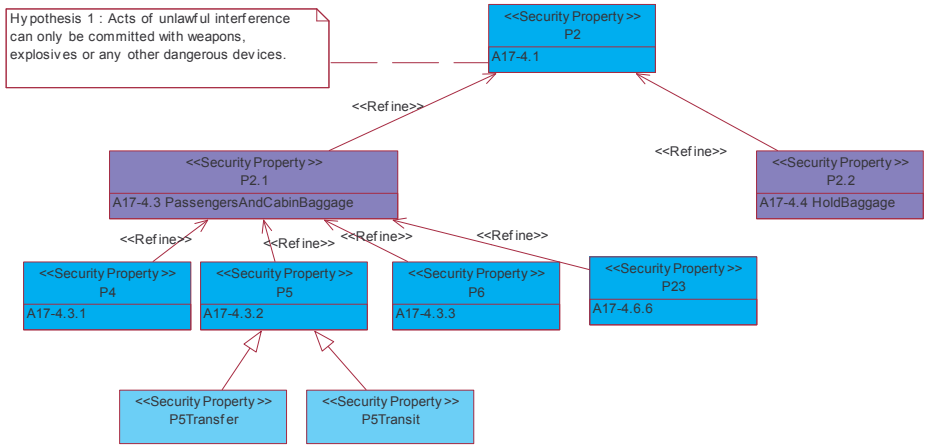

Figure 2: Extract of the security properties graph

Annex 17 distinguishes between six different situations that can lead dangerous objects to be introduced on board an aircraft (passengers and their cabin baggage, hold luggage, ground persons who can access the aircraft, ...) and specifies a list of numbered measures for each situation. This structuring has helped us to elaborate a tree that decomposes property *P2* into more specific sub-properties, so that a leaf level sub-property corresponds to an elementary numbered measure. We have identified 22 leaf level sub-properties. At a given level in the tree, a property is related to a set of sub-properties by a refinement relation. That means that satisfying all the sub-properties is a sufficient condition for satisfying the property. This tree can be considered as a special kind of NFR graphs (Mylopoulos et al., 1992; Chung, 1993) which are "and/or" graphs that decompose non functional requirements from vague abstractions to more concrete descriptions. Our tree is only a "and" graph that considers only security requirements.

From a graphical point of view, we have used a UML class diagram to describe the tree, to have a uniform representation, whatever the model. Each security property is modelled by a class, labelled by the stereotype `<<SecurityProperty>>` and the causal relations are expressed by navigable associations, labelled by the stereotype `<<Refine>>`. To ensure traceability between the diagram and the document, an attribute is assigned to each security property class: it refers to the paragraph or section of Annex 17 where the property is stated.

For example, Fig.2 considers the sub-diagram modelling the situation concerning passengers and their cabin baggage and corresponding to property *P2.1*. It is refined by four elementary properties. We detail the first two ones, properties *P4* and *P5*, because they will be used in the following sections of the paper. Security property P4 refers to paragraph 4.3.1 of Annex 17 whereas *P5* refers to paragraph 4.3.2.

P4: Originating passengers and their cabin baggage must be screened prior to boarding an aircraft.

P5: Transfer and transit passengers and their cabin baggage are subjected to adequate security controls prior to boarding an aircraft.

This property has been decomposed into two sub-properties *P5Transfer* and *P5Transit* which allow us to be more precise in the elaboration of the domain model (see next section).

In conclusion, we can say that the specific architecture of the standard has facilitated the properties identification step so that the obtained tree practically reflects the structuring of Annex 17, which will certainly make easier the validation of our models with the certification authorities. In Section 5, we present the main problems we have encountered during this step.

4.2 Domain Model of the Security Properties

Once the security properties have been identified, we elaborate a domain model for each of them. It specifies the objects, their relationships and their attributes which are relevant to formulate a security property. An object can be an entity or an agent. An agent is an active component, such as a human or an organization, that plays a role towards the satisfaction of some security properties.

For example, Fig.3 describes the objects derived from properties *P4* and *P5*. They are represented by the UML light-filled classes. The `ordinaryPassenger` class represents all the usual passengers[2] `embarked` in an `aircraftCabin` of an `aircraft`. A passenger can own (association `own`) `cabinLuggage`. A `cabinLuggage` can be carried by a passenger but also by another cabin person (a crew member or a pilot). Three kinds of passengers (`orginatingPassenger`, `transitPassenger`, `transferPassenger`) are distinguished in order to be more faithful to the properties. The model shows that originating passengers have a boolean attribute `screened`, while transit and transfer passengers have a `controlled` attribute. *P4* and *P5* require that these attributes are set to true for passengers embarked in an aircraft. This could be easily expressed using OCL (Warmer and Kleppe, 1999).

As far as possible, security concerns are integrated in the domain model in the following way. Even if a security property involves a number of domain objects, there is always a single object which plays a central role and which is the target of the security property. It is called subject matter or topic in (Cysneiros and Leite, 2004). For instance, the target of *P4* and *P5* is an aircraft. This object is depicted in the UML class diagrams as a stereotyped class `<<securityTarget>>` and linked to the class which it is an instance of. By propagating the instantiation mechanism on the domain model from this object, we obtain a snapshot that determines all the objects affected by a given security property. The `<<securityTarget>> aircraft` defines an `aircraftCabin`, a set of `ordinaryPassengers` with their `cabinLuggage`. In Annex 17, the security target is generally an aircraft, but it can

[2] There exist special categories of passengers such as passengers obliged to travel because of judicial of administrative reasons, law enforcement officers, ...

be for example the storage area of an airport, where mishandled baggage is kept. In this context, a security property should be linked to the more relevant objects of the snapshot defined by its target. Depending on whether the security property concerns objects of a same class or objects of different classes linked by an association, we define an association or an association class, stereotyped by `<<affect>>`. In Fig.3, the *P4* security property class is linked to the association `own` defined between the classes `originatingPassenger` and `cabinLuggage`.

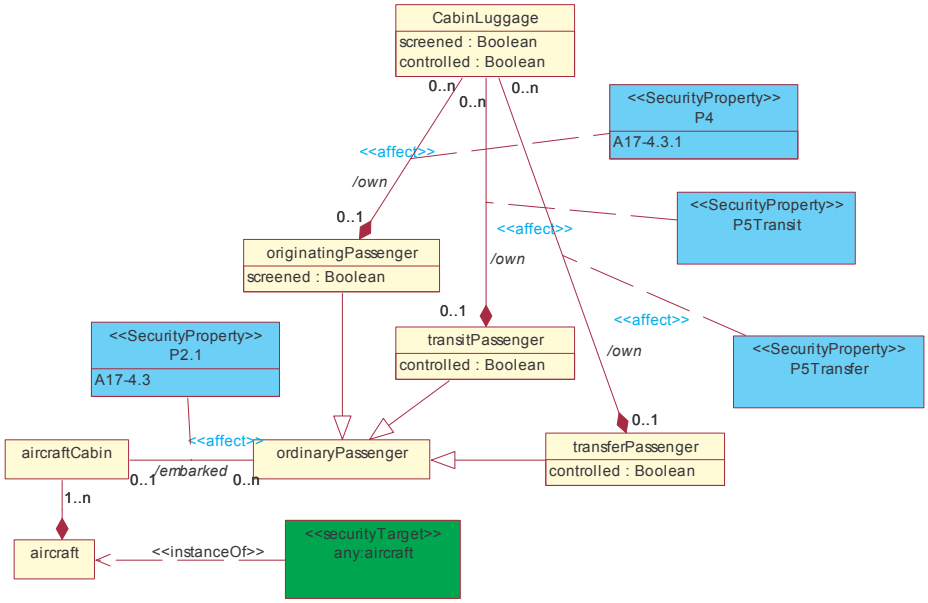

Figure 3: UML diagram for passenger checks

Generally in RE methods (Letier and Lamsweerde, 2002), an agent model captures responsibility links between agents and objects. Up to now, we did not feel a need to model agents apart from the domain model, because the very nature of Annex 17 is to define the obligations of each State member of the ICAO. Thus there is a single legislative agent, a State, who is legally responsible for the application of all the security properties. However, some articles refer to another kind of agent, aircraft operators, and their obligations, but it still remains under the responsibility of the State from which aircraft operators provide services. For example, in paragraph 4.4.3, one can read:

> *4.4.3 Each Contracting State shall establish measures to ensure that operators when providing service from that State do not transport the baggage of passengers who are not on board the aircraft unless that baggage is subjected to appropriate security controls which may include screening.*

The corresponding security property is:

P10: There is no baggage in the baggage hold whose owner is not on board the aircraft unless that baggage is controlled and possibly screened.

Figure 4 describes the domain model of property *P10*. The class `aloneHoldLuggage` contains hold luggage that do not belong to anybody, which implies a constraint on the association `HL_ownership`. The security target of the property is an aircraft that determines an aircraft operator. Again, we have introduced two new kinds of stereotyped class to model the agents: `<<responsibleAuthority>>` and `<<responsibleOrganization>>`, as it is shown in Fig.4. For the security property *P10*, an `aircraftOperator` plays the role of `<<responsibleOrganization>>` (stereotype `<<role>>`) who is responsible for the application of the security property (stereotype `<<application>>`), under the control (stereotype `<<control>>`) of a `contractingState` who remains the `<<responsibleAuthority>>`.

It is sure that this agent model will be significantly extended when we will take into account the other standard documents, and that, in particular, we will introduce "operational" agents who carry out the procedures established to implement the security properties.

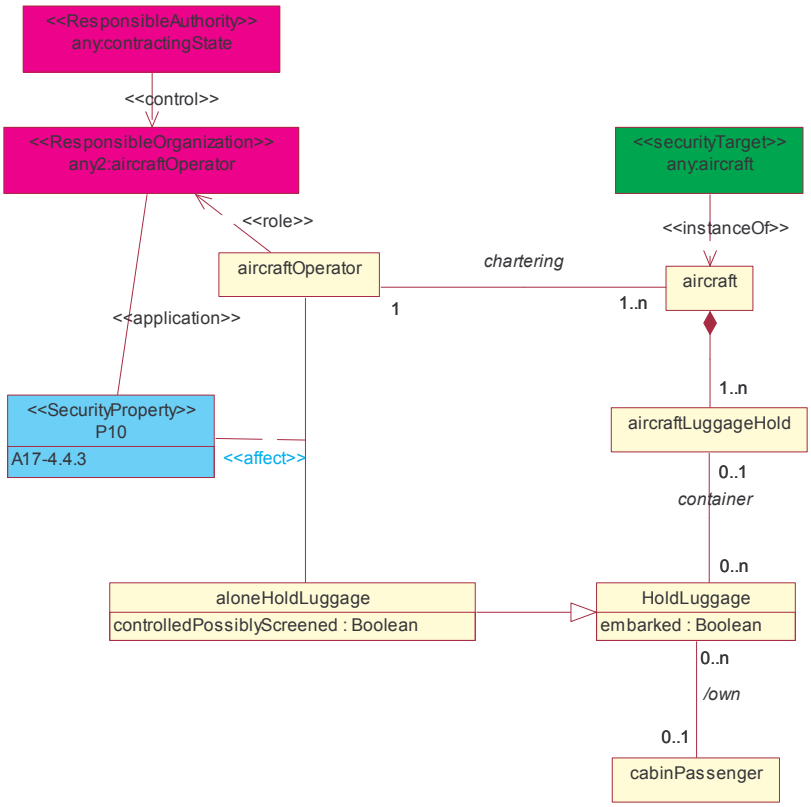

Figure 4: UML diagram for different kinds of agent

4.3 Conceptual Model of the Regulation Context

In parallel with the elaboration of the security requirements model, we need to model all the elements (physical characteristics, configuration, material, personal) and the procedures, the application of which is necessary for the safety or regularity of international air navigation. They constitute what we can call the existing physical system. More precisely, we consider different levels of the existing system. At the Annex 17 level, the system is obviously abstract: it corresponds to an abstract airport with the characteristics and the functionalities described in Annex 17. On the other hand, at the airport implementation level, the system is completely concrete: it takes into account the specificities of each airport, such as the actual buildings configuration, the number of ground agents or the number of screening points and their layout. The abstract model of the system will be used to check that Annex 17 is consistent, i.e. that the described procedures are conformed to the security properties.

The static aspect of the system is specified using a UML class diagram. It is obtained by merging all the class diagrams of the security properties and by incorporating additional information that comes from other chapters of Annex 17. Figure 5 is an extract that represents all the objects which can be in an aircraft and to which they can belong to. An `object` that is inside an `aircraft` can be either in an `HoldLuggage` or a `cabinLuggage` or carried by a `cabinPerson`. In addition we have defined integrity constraints because of the existence of association cycles.

Procedures are defined as operations in the relevant classes. At this abstract level, there are only two categories of operations: loading a luggage in the hold of an aircraft and loading someone in the cabin. For instance, operation `loadInCabin` of the class `cabinPerson` is described in natural language as:

loadInCabin(AC : aircraftCabin, CP : cabinPerson)
precondition : *if there are dangerous objects inside the cabin luggage of CP or carried by CP then they are authorized*
postcondition : *CP is loaded in AC (embarked = true)*

This operation is further refined to take into account the different kinds of cabin persons and the different procedures which they are subjected to in order to embark in an aircraft cabin.

To improve traceability with the domain models of the security properties, we have associated to each operation the security property that it must establish. For instance, the operation `loadInCabin` of the class `cabinPerson` must establish property *P2*.

In conclusion of section 4, existing requirements engineering approaches have up to now proved to be rather well adapted to our project. We have defined new elements or specialized existing ones to take into account the characteristics of airport security requirements, such as the notions of security target, responsible authority and responsible organization. These concepts have been introduced in UML diagrams thanks to the definition of stereotypes.

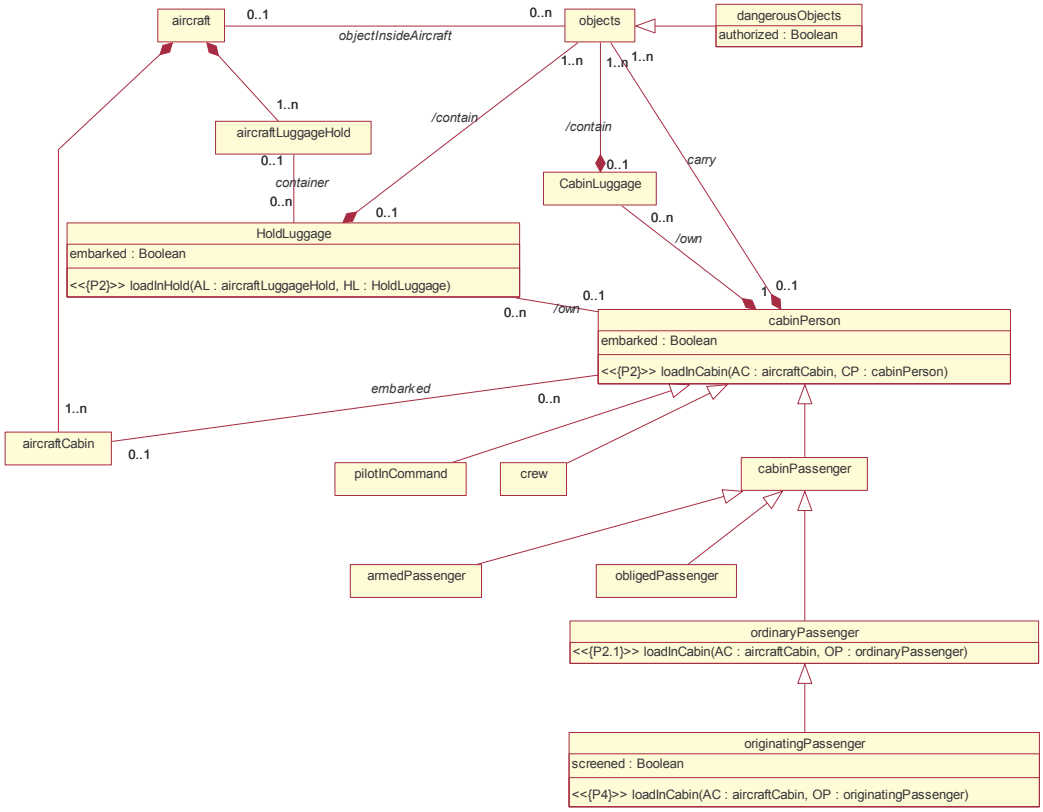

Figure 5: UML diagram for an abstract airport (extract)

A first version of the requirements and system models is achieved and has been validated by the members of the project who belong to the ONERA[3] laboratory. On the other hand, the ICAO has validated the process that we have defined and that is described in Fig.1. Indeed, experts are actually able to read and understand easily the different UML diagrams, without being obliged to understand the formal models. The next step is to validate the models by the ICAO and the ECAC.

5 Sample Results

Constructing the requirements and system models leads to a deep study of the standard and often reveals potential inconsistencies. In this section, we highlight some problems we have encountered and how, even without formalizing completely the standard, a first partial formalization can help to solve them. We focus on the notion of *authorization* which has a significant importance in regulation texts.

[3] ONERA : French Aeronautics and Space Research Center

5.1 Ambiguity of natural language

Paragraph 4.1 (see its wording page 6) is the main paragraph of chapter 4 because it states what are the objectives of the preventive security measures. However it appears to be ambiguous.

The point is about interpretation of the rider "the carriage or bearing of which is not authorized". That could mean either "the carriage or bearing of weapons, explosives or any other dangerous devices is **neither** authorized", or "weapons, explosives or other dangerous devices may not be introduced on board an aircraft **unless** their carriage or bearing is authorized". It is not clear at all which is the good interpretation, even if the latter seems more right than the former. We have carried out a survey among a group of native speakers of English that confirms the ambiguity of paragraph 4.1. Most of them declared that it is the context of the sentence that allowed them to make a distinction between the two interpretations and to choose the second one. The French translation is more explicit, because it introduces a conjunction "et" saying that the authorization is a supplementary condition for the carriage or bearing of weapons, explosives or other dangerous devices.

Formalization of both interpretations of paragraph 4.1 helps to understand the different meanings. We assume that there is a set of all the *objects*, and three subsets: *weapons*, *explosives* and *dangerousDevices*. For the second interpretation, we also need an attribute *authorization* which indicates if an object is given an authorization or not. The first interpretation is defined by the following subsets:

$$unauthorized\_carriage_1 = weapons \cup explosives \cup dangerousDevices$$

while the second one is defined by:

$$unauthorized\_carriage_2 = \{object \mid object \in weapons \cup explosives \cup dangerousDevices \wedge authorization(object) = no\}$$

Because the notation is formal, a question immediately arises about the domain of function *authorization*. The domain could be the set of all the objects or more specifically the set of weapons, explosives and dangerous devices. It seems to us more realistic to choose the second case, so:

$$dom(authorization) = weapons \cup explosives \cup dangerousDevices$$

Questioned about this ambiguity, the ICAO answered that it agrees with the problem but that the second interpretation is of course the correct one. This is determined by the context of the regulation. In fact, it appears that the regulation is not a close world and that the general context must be taken into account.

5.2 Multiple meanings of authorization

Chapter 4 is only composed of twenty-height paragraphs, and the word "authorized" (or unauthorized) is used inside nine of them. Moreover, it qualifies various names, such as "carriage or bearing is not authorized", "unauthorized persons", "unauthorized articles", "hold baggage authorized for carriage", "unauthorized interference", "authorized person", "special authorization". A problem is then to determine if every occurrence refers to the same meaning, or if some differences can happen. As an example, we propose to focus the exercise on the notion of objects which are

authorized or not authorized for carriage on board aircraft. The same exercise could be done for the persons.

To enlarge the context, let us take other paragraphs of this chapter that use the notion of "authorization". For example, in paragraph 4.3.2, one can read:

> *4.3.2 Each Contracting State shall ensure that transfer and transit passengers and their cabin baggage are subjected to adequate security controls to prevent unauthorized articles from being taken on board aircraft engaged in international civil aviation operations.*

What exactly does *unauthorized articles* mean? First, notice that "unauthorized" qualifies "articles", while in item 4.1, it qualifies "carriage or bearing of weapons, explosives or any dangerous devices". So, the questions are: is the word "articles" roughly equivalent to "weapons, explosives or any dangerous devices"? is the word "articles" equivalent to any "objects"?

A step further, paragraph 4.4.7 is written with another form:

> *4.4.7 Each Contracting State shall establish measures to ensure that aircraft operators when providing a passenger service from the State transport only hold baggage which is authorized for carriage in accordance with the requirements specified in the national civil aviation security programme.*

This paragraph speaks about "authorized hold baggage". One can infer that "hold baggage", or their content, is of the same nature as "articles". Here again, the point is the use of "authorized". We cannot expect that the meaning is the one used in the second interpretation of the first paragraph because in that case, an aircraft can only transport *authorized* weapons, explosives or other dangerous devices. So, clearly, the word "authorized" means something larger, probably a set containing the ordinary objects and the authorized (and dangerous) objects above.

Again, we present the interest of formalizing. For paragraph 4.3.2, the question is again about the domain of the authorization function. It is reasonable to consider that unauthorized articles are the same as the objects, the carriage or bearing of which is not authorized. Following the second interpretation of 4.1, we take the set below for the unauthorized articles:

$unauthorized\_articles = unauthorized\_carriage_2$

For paragraph 4.4.7, we have to determine the meaning of word "authorized" with respect to the one of "unauthorized", already formalized. As informally guessed, it is not the set of weapons, explosives or any dangerous devices where the function "authorization" returns "yes". The notion of "authorized hold baggage" should be here the complement of *unauthorized_articles* with respect to the set of all the objects, that is to say:

$$authorized\_hold\_baggage = \{object \mid (object \in objects \wedge object \notin weapons \cup explosives \cup dangerousDevices) \vee (object \in weapons \cup explosives \cup dangerousDevices \wedge authorization(object) = yes)\}$$

Two conclusions can be drawn from the partial formalization described in this section. First, a formal specification leads to ask very precise questions about the exact meaning of words that even the ICAO and ECAC specialists have not thought of. Second, some words appear to be ambiguous in the text but we need to take into account the general context of the regulation to remove this ambiguity. Thus, validation with the certification authorities is necessary.

6 Conclusion

The 9/11 attacks (9/11 report, 2004) have revealed the need for enhanced civil aviation security. One of the ways to improve security is to build on high quality standards. The most important results of the paper are twofold. First we have shown that existing RE methods, mainly dedicated to build software, can be applied to model standard. We have adopted a goal-oriented method, composed of three steps. The goals described in the standards are exclusively security properties. The first step consists in building the hierarchy of properties based on a refinement relation. Then a domain model is elaborated for each property. Finally, agents responsible for the application of security properties are determined. The second result relies on the definition of a set of extensions to the classical goal-oriented RE method to take into account our specific situation, more precisely the characteristics of airport security requirements. We have used UML notations as the representational view of our RE models, and the new concepts are naturally introduced by the definition of stereotypes. We have also shown on a typical example how formal notations can help ambiguities and incompleteness of the standard to be solved.

We are currently studying the complete formalization of Annex 17 by using different formal methods and we also plan to consider the other standards in order to detect conflicts between standards. SAFEE (Safee, 2004) is another project, funded by EU's Sixth Framework Programme for RTD (FP6), which aims to apply similar methods to security but during the flight. However, we do not have more details since up to now, no public papers are available. Finally, we believe that our approach can be applied to several application domains in the aeronautics industry, or more generally in any transportation industry where security or safety are built on international standards.

References

9/11 report (2004). The 9/11 Commission Report - Final Report of the National Commission on Terrorist Attacks Upon the United States. http://www.gpoaccess.gov/911/.

A17 (2002). Annex 17 to the Convention on International Civil Aviation - Security - Safeguarding International Cicil Aviation against acts of unlawful interference.

J.R. Abrial (1996). *The B-Book*. Cambridge University Press.

F. Bouquet, B. Legeard and F. Peureux (2004). A constraint solver to animate a B specification. *Int. Journal on Software Tools for Technology Transfer, Springer Verlag*, Vol. 6.

O. Carton (2004). *Modélisation de la sûreté dans le transport aérien*. Master thesis CNAM, ONERA, Toulouse, France.

L. Chung (1993). Dealing with Security Requirements During the Development of Information Systems. *Proceedings CAiSE '93, 5th Int. Conf. Advanced Information Systems Engineering, Lecture Notes In Computer Science*, Vol. 685, Paris, France.

Cysneiros,L.M. and Leite, J.C.S.P. (2004). Non-Functional Requirements: From Elicitation to Conceptual Model. *IEEE Transactions on Software Engineering*, vol.30, no5, pp. 328—350.

Dardenne, A., Lamsweerde, A. v., and Fickas S. (1993). Goal-Directed Requirements Acquisition. *Science of Computer Programming*, vol.20, pp. 3--50.

S. Dupuy, Y. Ledru, and M. Chabre-Peccoud (2000). An Overview of RoZ : a Tool for Integrating UML and Z Specifications. *Proceedings CAiSE'2000, 12th Int. Conf. on Advanced Information Systems Engineering, Lecture Notes In Computer Science*, Vol. 1789, Paris, France.

Edemoi (2004). EDEMOI project web site, http://www-lsr.imag.fr/EDEMOI/.

R. Laleau and A. Mammar (2000). An Automatic Generation of B Specifications from Precise UML Notations for Data Intensive Applications. *Proceedings ASE'2000 : International IEEE Conference on Automated Software Engineering*, Grenoble, France.

Axel van Lamsweerde (2000). Formal specification: a roadmap. *In A.Finkelstein, editor, ICSE - Future of SE Track*, pages 147--159, ACM Press.

Y. Ledru (1997). Specification and animation of a bank transfer using KIDS/VDM. *Automated Software Engineering*, vol.4 (1) pp. 33--51.

E. Letier and A. van Lamsweerde (2002). Deriving Operational Software Specifications from System Goals. *Proceedings FSE'10 - 10th ACM S1GSOFT Symp. on the Foundations of Software Engineering*, Charleston, USA.

N. Lopez, M. Simonot and V. Viguié Donzeau-Gouge (2002). A methodological process for the design of a large system: two industrial case-studies. *Proceedings FMICS'02 - Formal Methods for Industrial Critical System*s.

J. Mylopoulos, L. Chung, E. Yu, and B. Nixon (1992). Representing and Using Non-Functional Requirements: A Process-Oriented Approach. *IEEE Transactions on Software Engineering*, vol.18, no.6, pp. 483--497.

Bashar Nuseibeh and Steve Easterbrook (2000). Requirements Engineering: a roadmap. *In A.Finkelstein, editor, ICSE - Future of SE Track*, pages 35--46. ACM Press.

SAFEE (2004). SAFEE targets on-aircraft security.
http://europa.eu.int/comm/research/aeronautics/info/news/article_681_en.html.

J. Spivey (1992). *The Z notation - a reference manual (second edition)*. Prentice Hall.

E. Yu (1997). Modeling and Reasoning Support for Early-Phase Requirements Engineering. *Proceedings of the 3rd IEEE International Symposium on Requirements Engineering (RE'97)*. Jan. 6-8, Washington D.C., USA. pp.226-235.

J. Warmer and A. Kleppe (1999). *The Object Constraint Language*. Addison-Wesley.