Formal Modeling of Airport Security Regulations using the Focal Environment

David Delahaye
CEDRIC/CNAM, Paris, France
David.Delahaye@cnam.fr

Jean-Frédéric Étienne
CEDRIC/CNAM, Paris, France
etiennje@cnam.fr

Véronique Viguié Donzeau-Gouge
CEDRIC/CNAM, Paris, France
donzeau@cnam.fr

Abstract

We present the formalization of regulations intended to ensure airport security in the framework of civil aviation. In particular, we describe the formal models of two standards, one at the international level and the other at the European level. These models are expressed using the Focal environment, which is an object-oriented specification and proof system. In addition, we show that these models are correct and complete thanks to the Zenon automated theorem prover, which is the dedicated reasoning support of Focal. Finally, we propose an automatic transformation of Focal specifications to UML class diagrams, in order to provide a graphical documentation of formal models for developers, and in the long-term, for certification authorities.

1 Introduction

Many human activities are controlled by regulations and standards. Regulations can be seen as a set of rules/specifications and a key element to guarantee their effective enforcement is to assess the conformity of the procedures and artifacts they intend to regulate. However, the conformity assessment procedures are worthless if the correctness, completeness and consistency of the specifications are not established. Standards and recommended practices are usually written in natural language in order to be easily understood and adopted by a large number of stake-holders. Nevertheless, the normative documents are generally of voluminous size, ambiguous and often open to interpretation. Moreover, it is very difficult to automatically process natural language documents in search for inconsistencies. All these problems highlight the lack of a formal drafting process and this is where modeling techniques can help. Recent work [4] has shown that there is an increased interest in providing automated and systematic support to reason about regulations due to the growing complexity of safety and security requirements.

In this paper, we report on our experience of a 4 year study, which consists in building and analyzing the formal models of two standards related to airport security: the first one is the international standard Annex 17 [8], produced by the International Civil Aviation Organization (ICAO), an agency of the United Nations; the second one is the European Directive Doc 2320 [6], produced by the European Civil Aviation Conference (ECAC) and which is supposed to refine the Annex 17 at the European level. This formalization was realized using the Focal [7] environment, within the framework of the EDEMOI[1] [5] project. The EDEMOI project aims to integrate and apply several requirements engineering and formal methods techniques to analyze regulations in the domain of airport security.

In this project, we achieved several contributions. First, the formalization of the two standards (previously mentioned) allowed us to improve the quality of the normative documents and hence to increase the efficiency of the conformity assessment procedure. Second, thanks to this significant formalization, it was possible to validate the design features as well as the reasoning support offered by Focal. The specification environment was also extended to provide an appropriate level of documentation for the formal models. This extension mainly supports the production of a graphical documentation for Focal specifications in the form of UML class diagrams. The documentation is intended to be used by developers, and in the long-term, to facilitate discussions with certification authorities.

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

This paper is organized as follows: first, Section 2 provides a quick overview of the Focal environment; next, Sections 3 and 4 present respectively the formal models of the two standards seen previously and their corresponding validation; Section 5 describes the extension of Focal which allows us to produce UML models for documentation; finally, Section 6 summarizes the lessons that could be drawn from the analysis, formalization and validation of the regulations considered.

2 The Focal Environment

Focal [7], initiated by T. Hardin and R. Rioboo, is a language in which it is possible to build certified applications step by step, going from abstract specifications, called *species*, to concrete implementations, called *collections*. In this language, the first major notion is the structure of species, which corresponds to the highest level of abstraction in a specification and which has the following syntax:

```
species <name> =

  rep [= <type>];              (* abstract/concrete
                                  representation *)

  sig <name> in <type>;        (* declaration *)
  let <name> = <body>;         (* definition *)

  property <name> : <prop>;    (* property *)
  theorem <name> : <prop>      (* theorem *)
  proof : <proof>;

end
```

where <name> is simply a given name, <type> a type expression, <body> a function body, <prop> a (first-order) proposition and <proof> a proof.

Species can be combined using (multiple) inheritance (which works as expected) and can be parameterized either by other species or by entities from species. These two features complete the previous syntax definition as follows:

```
species <name> (<name> is <name>,
                <name> in <name>, ...)
        inherits <name>, <name> (<pars>),
                 ... = ... end
```

where <pars> is a list of <name>, which denotes the names used as effective parameters. When the parameter is a species parameter declaration, the "is" keyword is used. When it is an entity parameter declaration, the "in" keyword is used.

The other main notion of the Focal language is the structure of collection, which corresponds to the implementation of a specification (every attribute must be concrete). The syntax of a collection is the following:

```
collection <name> implements <name>
           (<pars>) = ... end
```

The certification of a Focal specification is ensured by the possibility of proving properties using Zenon, a first-order automated theorem prover, which is the reasoning support of Focal.

For further information regarding Focal and, in particular, for examples of specifications, the reader can refer to [1, 7], as well as to the Focal Web site[2].

3 Formal Modeling

This section presents the formal models realized in Focal for the two standards considered. The entire formalization takes about 10,000 lines of Focal code, with in particular, 150 species and 200 proofs. These models are more extensively described in [1].

3.1 Annex 17

In regulation modeling, it is important for the formal models to impose a certain structure that facilitates the traceability and maintainability of the normative documents. To achieve this purpose, an analysis is performed to organize the regulation into an hierarchy of goals. On one hand, the fundamental security properties are identified and are decomposed into sub-properties. On the other hand, a bottom-up approach is considered to determine how the sub-properties intend to satisfy the fundamental ones. In so doing, this may unveil any implicit hypotheses that led to the formulation of the preventive security properties. We also advocate that by reasoning on the hierarchy obtained, we can determine the correctness and completeness of the regulation.

The formal model presented in this subsection is structured according to the hierarchy of goals established. Each category of prevention, described in Chapter 4 of Annex 17 [8] (the international standard proposed by the ICAO), is represented by a Focal species named adequately to ensure traceability. For example, the security properties related to access control (A17, 4.2) are formalized in species a17property4_2. These species are specified by extending the portion of the domain environment they regulate while preserving the dependencies between the security properties. Moreover, to clearly make a distinction between the security requirements and the ways/means to implement them, the security properties are defined as invariants. In fact, from the formal model produced, it should be possible to rigorously assess the conformity of the security procedures implemented by each airport security programme.

[2] http://focal.inria.fr/.

Each category of prevention is also accompanied by appropriate correctness and completeness theorems, which aim to establish the derivability of the security property decompositions involved. Examples of correctness and completeness theorems are given in Section 4. Finally, the overall validation of Annex 17 is established in species annex17, where the fundamental security property defined in paragraph 4.1 of Annex 17 is specified. The general structure of the Annex 17 model is represented by Figure 1, where nodes are species and arrows are inheritance relations such that $A \leftarrow B$ means species B inherits from species A.

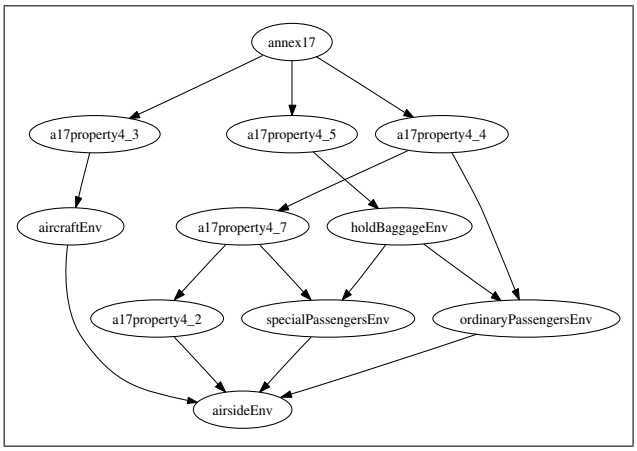

Figure 1. Structure of Annex 17

3.2 Doc 2320

The document structure of Doc 2320 [6] (the European standard proposed by the ECAC) is mainly organized according to the different categories of prevention described in Chapter 4 of Annex 17. Refinement in Doc 2320 appears at two levels. At the subject level, the refinement consists in enriching the characteristics of the existing subjects or in adding new subjects. At the security property level, either new security measures are introduced to sustain some specific security objectives, or each existing Annex 17 security measure is made more precise and sometimes more restrictive. The correctness and completeness of Doc 2320 are determined in the same way as for Annex 17. However, since Doc 2320 refines Annex 17, an additional verification is required to show that the security measures it describes do not invalidate (or are not less restrictive than) the ones defined in Annex 17. Thus, in addition to correctness and completeness proofs, another kind of proofs appears, that are refinement proofs (see Section 4). The model structure obtained for Doc 2320 is shown in Figure 2, where the Focal model corresponding to Annex 17 is represented with dashed nodes.

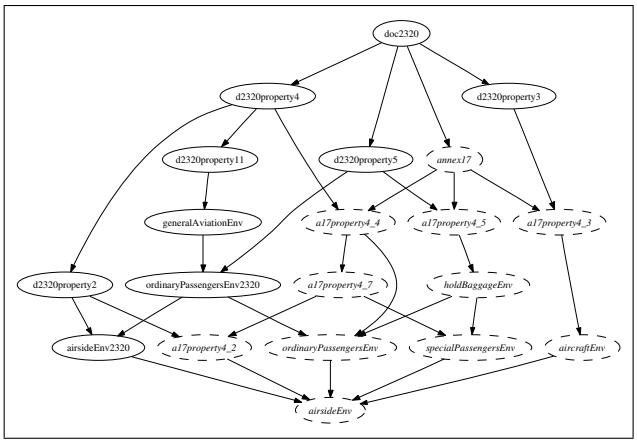

Figure 2. Structure of Doc 2320

4 Validation

In this section, we present the different analyses performed on the formal models produced in order to establish the correctness and completeness of the Annex 17 and Doc 2320 standards. The corresponding theorems are proved using Zenon. For more details, the reader can refer to [2].

4.1 Correctness and Completeness

As stated previously, we may assess the extent to which the regulation is complete by providing a formal proof for each security property decomposition obtained. In so doing, we may reveal either that the regulation conveys sufficient details to establish each causality relationship, or that some additional assumptions are required for the corresponding correctness proofs to be successful. In this context, by correctness, we mean that the preventive security measures are sufficient to satisfy the fundamental ones, i.e. a fundamental security property is implied by its sub-properties; by completeness, we mean that the preventive security measures are necessary to establish the fundamental ones, i.e. a fundamental security property is no longer satisfied when one of its sub-properties is omitted.

Example 4.1 (Correctness Proof) As an example, we describe the correctness theorem established for Property 4.2, which regulates access control (A17, 4.2). This example puts in evidence that the Annex 17 regulation is, in a certain way, not correct since additional assumptions are required for the proof to be completed. To justify these hidden assumptions, we need to consider how some of the security properties are formalized in species a17property4_2:

Property 4.2.1. Property 4.2.1 specifies that access to security restricted areas must be controlled in order to prevent unauthorized entry. It is formalized in two steps to properly characterize the notion of unauthorized entry:

```
property property_4_2_1a :
  all area in sra, all s in self,
    sra_set!member
      (area, !securityRestrictedAreas (s)) →
    sra!access_controlled (area);

property property_4_2_1b :
  all area in sra, all s in self,
    sra_set!member
      (area, !securityRestrictedAreas (s)) →
    sra!access_controlled (area) →
    all p in a_subject,
      as_set!member
        (p, sra!airsideSubjects_in_sra (area)) →
      sra!access_authorized (p, area);
```

where p is an airside subject, area a security restricted area and s a particular instance of the regulation modeled by species a17property4_2.

Property 4.2.3. Property 4.2.3 states that the identity of all airside subjects must be verified before access is authorized to security restricted areas. It is formalized as follows:

```
property property_4_2_3 : all area in sra,
  all p in a_subject, all s in self,
    sra_set!member
      (area, !securityRestrictedAreas (s)) →
    sra!access_authorized (p, area) →
    a_subject!identityVerified (p);
```

where p, area and s are specified as for Property 4.2.1.

Property 4.2. From the analysis performed on Annex 17, it is established that Properties 4.2.1 to 4.2.6 may be sufficient to guarantee the satisfaction of Property 4.2. In our formalization, Property 4.2 is therefore specified as a theorem for which a proof is required:

```
theorem property_4_2 : all p in a_subject,
  all a in d_aircraft, all area in sra,
  all s in self, all o in obj,
    sra_set!member
      (area, !securityRestrictedAreas (s)) →
    dep_ac_set!member
      (a, sra!departingAircraft_in_sra (area)) →
    !is_unescorted_person_vehicle (p, s) →
    obj_set!member
      (o, a_subject!objects_carried (p)) →
    obj_set!member
      (o, d_aircraft!onboardObjects (a)) →
    d_aircraft!access_authorized (p, a)
    and !no_unauthorized_objects (o, s)
proof : ...;
```

This property states that if an object carried by an airside subject satisfying predicate is_unescorted_person_vehicle is introduced on board an aircraft departing from a security restricted area, then the airside subject has authorized access to such aircraft. Moreover, if the object is classified as dangerous then it is authorized (specified by predicate !no_unauthorized_objects).

When attempting to prove the above theorem, we discovered that the following assumptions have to be made for the proof to be successful:

1. Unescorted persons may access a departing aircraft if they have their identity verified and their background checked.
2. Unescorted persons are trustworthy and therefore are considered not to carry any unauthorized dangerous objects.

Similar assumptions are required for airside vehicles, but with subtle differences.

4.2 Doc 2320 : Refinement Theorems

As said in Section 3, since Doc 2320 is supposed to refine Annex 17 at the European level, there is a need to ensure that its security properties are not less restrictive than or do not invalidate their Annex 17 counterparts. In our context, refinement theorems are therefore correctness theorems, but with a more specific nature.

Example 4.2 (Refinement Proof) As an example of a more restrictive security property, we may consider the refinement theorem established for Property 4.2.6 of Annex 17. At the Annex 17 level, it is stated that only a portion of unescorted persons accessing security restricted areas has to be screened, while at the Doc 2320 level, screening is made compulsory for all personnel (D2320, 2.3(a)). In species d2320property2, Property 2.3(a) of Doc 2320 is formalized as follows:

```
property d2320_2_3a : all area in sra,
  all p in a_staff, all s in self,
    sra_set!member
      (area, !securityRestrictedAreas (s)) →
    sra!access_authorized (!upToAs (p), area) →
    a_staff!handSearched (p) or
    a_staff!walkedThroughMetalDetection (p);
```

where the following property completes the formalization by specifying that hand search and Walk-Through-Metal-Detection equipment are considered as screening methods:

```
property inv_screening : all s in self,
  !handSearched (s) or
  !walkedThroughMetalDetection (s) →
  !screened (s);
```

The corresponding refinement theorem is specified in species d2320property2 as follows:

```
theorem refinement_4_2_6 :
  !d2320_2_3a → !property_4_2_6
proof : ...;
```

5 From Focal to UML

This section presents an extension of Focal, which appeared quite necessary during our formalization and which aims to provide a graphical documentation of our formal models for developers. In the long term, the idea is to provide higher-level views that would be more pertinent to certification authorities. This extension consists of an automatic transformation of Focal specifications into UML class diagrams. The transformation is based on a formal description for a subset of the UML 2.1 static structure constructs. The UML metamodel is also tailored to consider the semantic specificities of the Focal specification language (we adopted a profile approach). The corresponding rules are described in details in [3]. In addition, this transformation was proved to be sound: the defined profile does not introduce any inconsistency w.r.t. the well-formedness rules of the UML metamodel; the UML model obtained from a well-typed Focal specification satisfies both the well-formedness rules of the UML metamodel and the constraints within the profile. Figure 3 shows the UML model obtained from an excerpt of the formalization realized for airport security regulations and which concerns cabin persons.

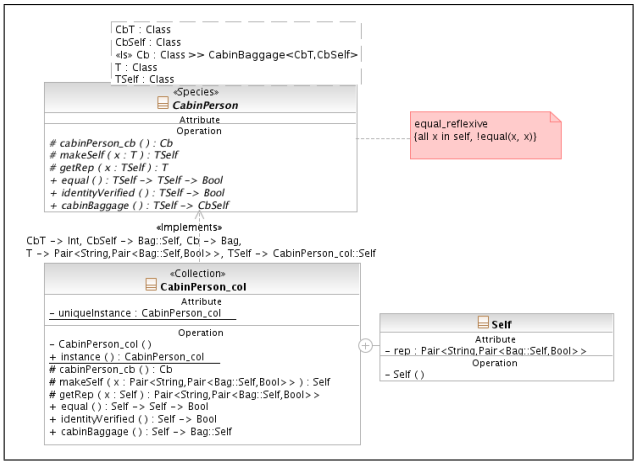

Figure 3. CabinPerson Classes

6 Lessons Learned

The simple fact of organizing the regulations into a hierarchy of goals helps have a better understanding of the airport security policy. The formalization of the Annex 17 and Doc 2320 mainly corresponds to a knowledge engineering task. In particular, to properly capture the meaning of the identified security properties, it is essential to provide an appropriate vocabulary, while preserving as far as possible the nomenclature of the normative documents. The formal modeling process has mainly served to clarify various ambiguities and imprecisions residing in the informal definitions of the security properties considered, hence improving the quality of the normative documents.

The correctness theorems proved during the validation step have served to clarify any remaining imprecision in the formal models. In essence, by systematically exploring the hierarchy of goals established, we managed to identify a set of hidden assumptions (or omissions) that shed light on the intention of some specific security properties. Regarding the refinement theorems, they allowed us to formally establish that the Doc 2320 regulation indeed refines Annex 17 at the European level. It should be noted that for some exceptional cases, we needed to adopt specific refinement validation patterns, which we here omit due to space restrictions. For instance, the omission or partial refinement of a higher-level security property at the Doc 2320 level is not necessarily considered as a deficiency of the regulatory system. It can be the case that the drafter may assume that any omitted higher-level requirement is considered to be left unchanged and is still applicable.

References

[1] D. Delahaye, J.-F. Étienne, and V. Viguié Donzeau-Gouge. Certifying Airport Security Regulations using the Focal Environment. In *Formal Methods (FM)*, volume 4085 of *LNCS*, pages 48–63. Springer, Aug. 2006.

[2] D. Delahaye, J.-F. Étienne, and V. Viguié Donzeau-Gouge. Reasoning about Airport Security Regulations using the Focal Environment. In *International Symposium on Leveraging Applications of Formal Methods, Verification and Validation (ISoLA)*, pages 45–52. IEEE CS Press, Nov. 2006.

[3] D. Delahaye, J.-F. Étienne, and V. Viguié Donzeau-Gouge. Producing UML Models from Focal Specifications: An Application to Airport Security Regulations. In *Theoretical Aspects of Software Engineering (TASE)*. IEEE CS Press, June 2008.

[4] R. Laleau and M. Lemoine, editors. *International Workshop on Regulations Modelling and their Validation and Verification (REMO2V), in conjunction with Conference on Advanced Information Systems Engineering (CAiSE)*. Presses Universitaires de Namur, June 2006.

[5] The EDEMOI Project, 2003. `http://www-lsr.imag.fr/EDEMOI/`.

[6] The European Civil Aviation Conference. *Regulation (EC) N° 2320/2002 of the European Parliament and of the Council of 16 December 2002 establishing Common Rules in the Field of Civil Aviation Security*, Dec. 2002.

[7] The Focal Development Team. *Focal, version 0.3.1*. CNAM/INRIA/LIP6, May 2005. Available at: `http://focal.inria.fr/`.

[8] The International Civil Aviation Organization. *Annex 17 to the Convention on International Civil Aviation, Security - Safeguarding International Civil Aviation against Acts of Unlawful Interference, Amendment 11*, Nov. 2005.