An attempt to combine UML and formal methods to model airport security

Yves Ledru[1], Régine Laleau[2], Michel Lemoine[3], Sylvie Vignes[4], Didier Bert[1], Véronique Donzeau-Gouge[5], Catherine Dubois[5], and Fabien Peureux[6]

[1] LSR/IMAG, Université de Grenoble, {Yves.Ledru, Didier.Bert}@imag.fr
[2] Laboratoire LACL, Université de Paris 12, laleau@univ-paris12.fr
[3] ONERA,Toulouse, lemoine@onera.fr
[4] GET/ENST, dépt Informatique et Réseaux, Paris, vignes@enst.fr
[5] Cedric/CNAM, Paris, donzeau@cnam.fr, dubois@iie.cnam.fr
[6] LIFC, Univ. de Franche-Comté, Besançon, peureux@lifc.fr

Abstract. The EDEMOI project aims to model standards that regulate airport security. It involves the production of a UML model, to support the validation activity, and a formal model for verification purposes. This paper discusses the use of the RoZ tool to establish a strong link between both models and lists the problems faced during this translation.

1 Introduction

A key element of aviation security is airport security, which prevents weapons and other dangerous objects from being brought on-board an airplane. Airport security is regulated by international standards such as Annex 17 of International Civil Aviation Organisation [4]. These are natural language documents with their usual problems: risk of ambiguity or incompleteness, and poor tool support for consistency checking and validation.

The goal of the EDEMOI project [1] is to model airport security on the basis of these international standards, and to check its consistency using the tools supporting formal methods such as B[2] or Z [5]. Fig. 1 gives the major documents of the project and the related stakeholders. There are two kinds of stakeholders. *Certification authorities* write the international standards and enforce their application in airports. In the EDEMOI project, they are expected to validate the models of these international standards. *Model engineers* produce the models and analyse their consistency, starting from the written international standards. The project involves three kinds of documents. *International standards* are natural language documents describing the rules of airport security. *Graphical models* are prepared by model engineers and validated by the certification authorities. *Formal models* are only accesssed by model engineers. Formal method tools (proof, test and animation tools) support their analysis.

The three kinds of documents are expected to describe the same reality. Therefore, links must be established between them. The links between the UML diagrams and the natural language documents are mainly informal and must be identified manually by model engineers. The links between the formal specifications and the UML diagrams should be automated since the validation of the formal models is only made indirectly by the certification authority, based on this link. This paper will focus on this link.

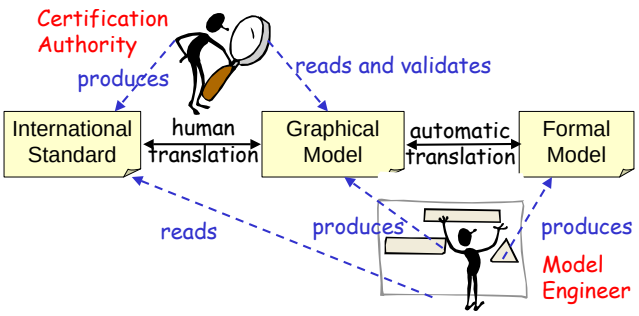

Fig. 1. The EDEMOI stakeholders

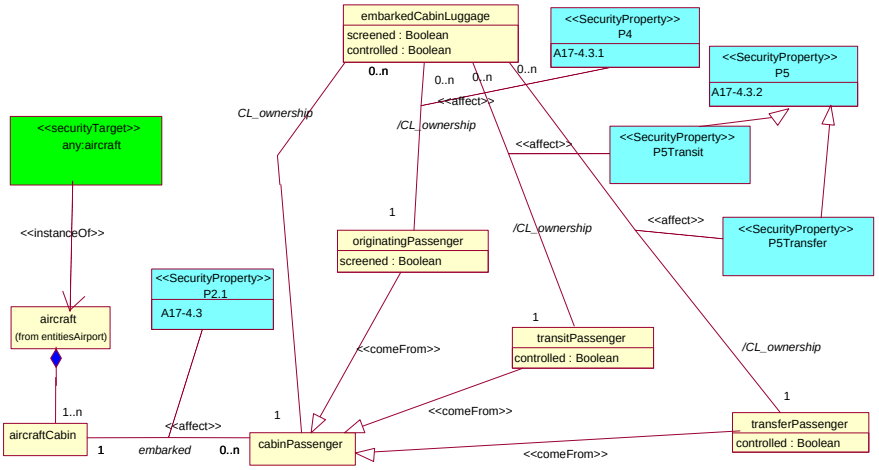

Fig. 2. Class diagram for passengers and cabin luggage with security properties

2 UML modeling

The translation from natural language to UML is a three-step process [6]: (1) identification of goals, i.e. security properties, (2) production of a domain model, (3) production of class diagrams which correspond to a subset of the domain model linked to the security properties. Our modeling process resulted in 19 diagrams (220 classes, 203 associations). Fig. 2 is one of these diagrams. It models passengers and their cabin luggage. Security properties appear as stereotyped classes linked to the classes and associations of the diagram. Another stereotype identifies the security target of the diagram (here the aircraft).

3 Translation into Z

Many properties that ensure airport security can be expressed as invariant properties. For example, the ultimate goal of airport security is to prevent dangerous objects from being brought on-board an airplane, expressed as the following invariant: *Objects that are on-board of an airplane are not dangerous.* In the passengers/luggage example, a similar property can be stated: *Passengers or*

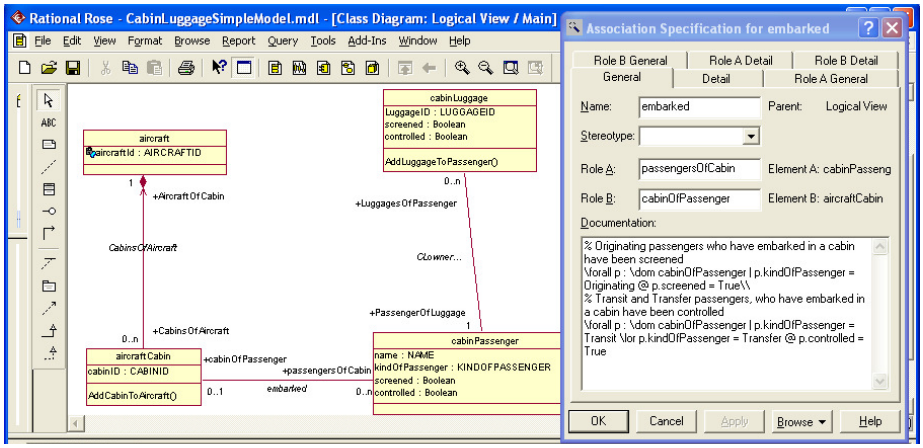

Fig. 3. A class diagram with Z annotations

cabin baggages embarked on an airplane have been screened or controlled. The choice of B and Z as target languages in the EDEMOI project corresponds to their adequacy to express such invariant properties, and to the availability of tools which support the translation from UML to formal methods.

RoZ is a tool that automates the translation of UML class diagrams with Z annotations, into a full Z specification [3]. The UML diagram defines the structure of the specification, while invariant properties are stated as supplementary annotations. The tool is actually a set of scripts on top of Rational Rose [7]. Z annotations are stored inside the documentation fields of the model, as shown in Fig. 3. RoZ allows annotations to be captured at several levels in the diagram (attribute level, class level, association level, diagram level).

The tool translates the annotated class diagram into a Z specification. It also supports the automatic generation of basic operations to create, delete and modify instances of the classes and relations. The translation was recently adapted to produce a specification that can be animated by the Jaza tool [8]. Snapshots of the animation are represented graphically as object diagrams. These animations and representations help understand and validate the specification.

Difficulties in translating UML into Z The class diagram of Fig. 2 had to be adapted in order to translate it with RoZ, as shown in Fig. 3. Many of the modifications are not specific to RoZ but would apply for any similar tool. The following modifications were performed.

- Security properties no longer appear as classes on the diagram.
- The security target has also been deleted.

Both cases correspond to stereotyped classes, which are simply translated as classes by RoZ. Since we don't want properties or security targets to be translated as classes in the Z model, these stereotyped classes were removed from the diagram and properties were included in the relevant documentation fields.

– Instead of using subclasses, the *kindOfPassenger* attribute has been added to the *cabinPassenger* class, and the attributes of subclasses (*screened* and *controlled*) have been added to the class.

RoZ supports the translation of inheritance, but the resulting specification can not be animated with Jaza. Therefore, we modified the diagram to fit the subset of UML that is translated into executable specifications.

– Every role of every association has a name.
– Identification attributes have been added to several classes (*cabinID*, *aircraftID*, *LuggageID*, *name*).

This is specific to RoZ. In RoZ, objects must have at least one attribute.

– The multiplicity of role *cabinOfPassenger* has been weakened from 1 to 0..1.
– The multiplicity of role *CabinsOfAircraft* has been weakened from $1..n$ to $0..n$.

Multiplicities in a class diagram put dynamic constraints on the execution of the model. Here, Fig. 2 requires to simultaneously create the aircraft and its first cabin. Weakening the constraint allows to first create the aircraft, and then the cabin.

4 Conclusion

This paper has presented the EDEMOI approach to model airport security. The model is based on two kinds of languages: graphical languages to support the validation activities, and formal languages to support verification. The existence of a strong link between graphical and formal models is mandatory to make sure that "what you validate is what you verify". This paper has presented the problems that arise when trying to translate graphical models into formal ones with the RoZ tool. Hopefully, several of these difficulties can be solved by extending RoZ. In particular, specific support can be designed for our stereotypes, and further work can result in a better support of inheritance.

Acknowledgment: This research is supported by the ACI Sécurité Informatique of the French Ministry of Research.

References

1. *EDEMOI project web site*, 2004. http://www-lsr.imag.fr/EDEMOI/.
2. J.R. Abrial. *The B-Book*. Cambridge University Press, 1996.
3. S. Dupuy, Y. Ledru, and M. Chabre-Peccoud. An Overview of RoZ : a Tool for Integrating UML and Z Specifications. In *12th Conference on Advanced information Systems Engineering-CAiSE'2000*, LNCS 1789. Springer, 2000.
4. ICAO. *Annex 17 to the Convention on Int. Civil Aviation - Security - Safeguarding International Civil Aviation against acts of unlawful interference*, 2002.
5. ISO. *Information technology – Z formal specification notation – Syntax, type system and semantics*, 2002.
6. R. Laleau, S. Vignes, Y. Ledru, M. Lemoine, D. Bert, V. Donzeau-Gouge, C. Dubois, and F. Peureux. Application of requirements engineering techniques to the analysis of civil aviation security standards. In *International Workshop on Situational Requirements Engineering Processes (SREP'05)*, 2005.
7. Rational Sofware. *Rational Rose*. http://www.rational.com.
8. Mark Utting. *The Jaza Animator*. http://www.cs.waikato.ac.nz/~marku/jaza/.