Safety Science 84 (2016) 225–237

Contents lists available at ScienceDirect

Safety Science

journal homepage: www.elsevier.com/locate/ssci

Hazard analysis of human–robot interactions with HAZOP–UML

Jérémie Guiochet

University of Toulouse, UPS, LAAS-CNRS, Toulouse, France

ARTICLE INFO

Article history:
Received 16 July 2015
Received in revised form 20 November 2015
Accepted 14 December 2015
Available online 11 January 2016

Keywords:
Hazard identification
Risk analysis
Robot safety
HAZOP
UML

ABSTRACT

New safety critical systems are about to appear in our everyday life: advanced robots able to interact with humans and perform tasks at home, in hospitals, or at work. A hazardous behavior of those systems, induced by failures or extreme environment conditions, may lead to catastrophic consequences. Well-known risk analysis methods used in other critical domains (e.g., avionics, nuclear, medical, transportation), have to be extended or adapted due to the non-deterministic behavior of those systems, evolving in unstructured environments. One major challenge is thus to develop methods that can be applied at the very beginning of the development process, to identify hazards induced by robot tasks and their interactions with humans. In this paper we present a method which is based on an adaptation of a hazard identification technique, HAZOP (Hazard Operability), coupled with a system description notation, UML (Unified Modeling Language). This systematic approach has been applied successfully in research projects, and is now applied by robot manufacturers. Some results of those studies are presented and discussed to explain the benefits and limits of our method.

1. Introduction

Besides the developments of well-known safety critical systems in aeronautics or transportation, new systems are about to appear in our everyday life: robots at home, at work, or in the hospitals (Royakkers and van Est, 2015). Such systems, will interact with users, and execute tasks in the vicinity or even in physical contact with humans. Hence, a failure of such complex systems may lead to catastrophic consequences for users which is a major obstacle to their deployment in real life. Most safety analysis techniques coming from the dependability (Avižienis et al., 2004) or risk management (ISO31000, 2009) domains could be used for such systems, but some specificities of robots limit their efficiency. For instance, the fact that robots move in unstructured and unknown environments makes the verification and validation (mainly through testing) non sufficient (it is impossible to guarantee that all main scenarios have been tested); the presence of users and complex non deterministic software (with decisional mechanisms) limit the use of quantitative risk analysis techniques; classical hazard analysis techniques are also not adapted to the complexity of human–robot interactions. Little work has been done about risk analysis for such systems, although it is a major challenge for robot certification (Mitka et al., 2012). Many robotics studies about estimation and treatment of collision risks exist (many references presented by Haddadin (2014)), but few are on risk analysis methods (Dogramadzi et al., 2014). The safety community has rarely addressed this issue, whereas we have been working on this for a decade (Guiochet and Vilchis, 2002; Guiochet et al., 2004).

Some robot manufacturers use directives (2006/42/EC, 2006) or standards (ISO13849-1, 2006) dedicated to machines, but they are not completely applicable, particularly when there is a human–robot physical interaction. Generic standards like IEC61508-5 (2010), are also hardly applicable due to uncertainties in the robot behavior (in this standard, fault correction through artificial intelligence is not recommended for safety integrity level SIL2 to SIL4). More recently, the standard ISO10218-1 (2011) for industrial robots that might share their workspace with humans, has been completed by the ISO13482 (2014). It is also important to note that such standards, do not cover other application domain robots. For instance, in the medical field, there is no robotic-specific standard, and the robots are considered as active medical devices such as defined in the 93/42/EEC (1993), and covered by ISO/FDIS14971 (2006) for risk management. In all those standards, classic risk management and design recommendations are proposed, but no specific guidelines for risk analysis techniques are presented.

E-mail address: jeremie.guiochet@laas.fr
URL: http://homepages.laas.fr/guiochet

http://dx.doi.org/10.1016/j.ssci.2015.12.017
0925-7535/© 2016 Elsevier Ltd. All rights reserved.

To cope with the previous issues, we suggest a hazard identification technique with the following objectives:

1. applicable from the very beginning of the development process,
2. includes human activity as a source of hazard,
3. provides guidance for analysts with list of guide words,
4. focuses on operational hazards, i.e., hazards linked with the robot tasks and interactions.

Among risk analysis techniques, the most widely used are Preliminary Hazard Analysis (PHA), Hazard Operability Analysis (HAZOP), Fault Tree Analysis (FTA), and Failure Mode, Effects, and Criticality Analysis (FMECA). The two first may be applied as hazard analysis at the very early steps of a development process, whereas FTA and FMECA are more dedicated to advanced steps, focusing more on reliability aspects. Thus, we chose to base our method on HAZOP, and to combine it with the system modeling language UML (Unified Modeling Language). This method developed at LAAS (Guiochet et al., 2010, 2013; Martin-Guillerez et al., 2010), has been successfully applied in several French and European projects (PHRIENDS, 2006–2009; SAPHARI, 2011–2015; MIRAS, 2009–2013) in collaboration with robot manufacturers (KUKA Robotics, AIRBUS Group and Robosoft). This paper synthesizes for the first time our work on HAZOP–UML, and proposes an analysis of the applications in these projects.

The remainder of this paper is structured as follows. Section 2 provides background on UML and HAZOP. In Section 3, we present the HAZOP–UML method, and in Section 4, results of several experiments are analyzed and discussed. In Section 5, related work on model-based safety analysis is compared to our approach. We conclude in Section 6 by outlining the benefits and limits of HAZOP–UML, and listing some future directions.

2. Background

2.1. Unified Modeling Language

UML (Unified Modeling Language) is a graphical notation, widely used in software and system engineering domains to support early steps of the development process. Its specification is available on the Object Management Group UML page.[1] The current version (UML 2), has thirteen diagrams, that could be classified in static diagrams (e.g., class diagram) and dynamic diagrams (e.g., use case, sequence and state machine diagrams). UML is a language, and not a method, as it is not specified in which chronological order each diagram must be used. But, use cases and sequence diagrams are typically used at the beginning of any project development. State machine diagrams are also widely used in reactive systems as robot controllers. Hence, we will present those three diagrams, focusing only in the elements we will use for our approach. One main pitfall using this language is to mix different levels of details in the same diagram. For instance, mixing some high level specifications with implementation constraints on the same diagram is error prone and also not recommended for the safety analysis. This is why we also put forward in this paper some modeling rules to avoid this pitfall and to guide the analysts.

As a running example, we will use some models of the case study MIRAS (2009–2013), an assistive robot presented Fig. 1, for standing up, sitting down and walking, and also capable of health-state monitoring of the patients. It is designed to be used in elderly care centers by people suffering from gait and orientation problems where a classic wheeled walker (or "rollator"), is not sufficient for patient autonomy. The robotic rollator is composed of a mobile base and a moving handlebar.

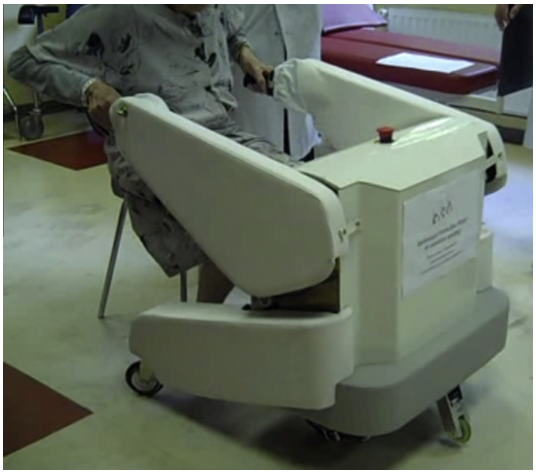

Fig. 1. MIRAS robot prototype during clinical investigation.

2.1.1. Use case diagrams

This diagram is the basic requirement UML model, presenting the system to analyze, the actors communicating with it, and the objectives for the use of the system: the use cases. The example of Fig. 2 only presents a subset of the complete use case diagram (15 use cases), and the two involved actors. In this diagram, the proposed services are to help the patient to stand up (UC02), deambulate (UC01), and sit down (UC03). The system is also able to detect physiological issues and trigger an alarm (patient heartbeat and fatigue, in UC08). We also represent that the system offers the profile learning facility (UC10). In some projects using UML the mechanical part of a robot is represented as a UML actor, and the system boundary (the box around use cases) defines the robot controller (including software and hardware). We do not recommend using such an approach to perform the hazard identification, indeed, the complete system has to be studied as a whole.

This diagram provides an expressive and simple mean to communicate between developers, analysts and users. This graphical representation is always completed with a textual description as in Fig. 3. Important information such pre and post conditions, and non-functional requirements are included. Use case diagram only represents functional requirements. Textual description of the normal, alternative and exception flows may also be presented with sequence diagrams as presented hereafter.

In the UML OMG standard, some relations may exist between use cases (mainly the relations *extend* and *include*) but we recommend not to use them, as they often lead to misunderstandings and to an unclear application of the HAZOP–UML method. In order to prepare the HAZOP–UML study, an extract from the use case textual description should be done, with only the pre and post conditions, and also the invariants coming from safety properties in the "Non functional requirements" category. An example of such a table is given in Fig. 4 for the UC02 of the MIRAS running example.

2.1.2. Sequence diagrams

Fig. 5 shows a sequence diagram, describing a possible scenario, which is actually an instance of an UML use case. This diagram shows a nominal scenario for the UC02. Other scenarios are possible for the UC02, like alternative flow of events (e.g., the patient releases the handles while she is standing up). This second scenario will be represented with another sequence diagram (not presented here). The expressiveness of such diagram is well adapted to represent human–robot interactions, and have proven to be useful while discussing with other stakeholders who are not experts in this language (doctors, mechanical engineers, etc.). All messages

[1] www.uml.org: accessed 2015-05-15.