A UML-based method for risk analysis of human-robot interactions

Damien Martin-Guillerez, Jérémie Guiochet, David Powell and Christophe Zanon
Université de Toulouse ; UPS, INSA, INP, ISAE ; LAAS ; F-31077 Toulouse, FRANCE
CNRS ; LAAS ; 7 avenue du colonel Roche, F-31077 Toulouse, FRANCE
firstname.lastname@laas.fr

ABSTRACT
Safety is a major concern for robots that interact physically with humans. We propose a risk analysis method based on deviation analysis of system usage scenarios that allows the identification of major risks. Scenarios are described with the common Unified Modeling Language (UML), and risk analysis is performed with the guideword-based collaborative method HAZOP (HAZard OP-erability). We adapt HAZOP attributes and guidewords for generic interpretation of UML use-case and sequence diagrams describing human-robot interactions. This approach has been systematically applied for the analysis of two quite different robots working in a human environment: a mobile manipulator and a robotic strolling assistant. When applied, the method gave conclusive evidence that the modeled systems were not safe. A CASE tool to support this method is also presented.

Categories and Subject Descriptors
K.4.1 [**Computers and Society**]: Public Policy Issues—*Human safety*; K.6.1 [**Management of Computing and Information Systems**]: Project and People Management—*Systems analysis and design*; D.2.2 [**Software Engineering**]: Design Tools and Techniques—*CASE*

General Terms
Design, Security, Human Factors

Keywords
Safety, risk analysis, scenario, HAZOP, UML

1. INTRODUCTION
Previously confined to purely industrial applications, robots are now starting to directly interact with humans: assistive robots, medical robots or even pet robots. Such interactions between humans and robots can lead to hazardous situations for the humans, especially for medical robots. Therefore, when designing a robot interacting with humans, methods to analyze the safety of the robot should be used.

Traditional methods to tackle safety like Fault Tree Analysis or Failure Modes, Effects and Criticality Analysis raise different concerns. First, they are unsuited to analyze physical interactions between human and robot. Second, multiple stakeholders using their own languages and models need to share the effort of risk analysis. Traditional methods thus often give rise to consistency errors and understanding problems. Third, risk analysis should start at the very first steps of the development process. We propose a method to address these issues based on two well-known techniques. The Unified Modeling Language (UML) is used to describe the interactions between humans and the robotic system. Risk analysis is then performed on this model with the guideword-based collaborative method HAZOP (HAZard OPerability).

This method is designed to be used at the early stages of system development. Human-robot interactions are first described using UML use-case and sequence diagrams. The main advantages of using UML are that it is now a *de facto* standard for system description, it is easily understandable by non-experts, and it is well adapted for early stages of development. UML has already been used with success to analyse the safety of medical robot applications [9]. This application showed that this subset of UML (i.e., use-case and sequence diagrams) is well-adapted to model physical human-robot interaction. The HAZOP method is then adapted and applied to each element of the UML model. HAZOP is also well-adapted to early development stages. It is easily understandable and enables a systematic analysis through the use of guidewords. Finally, we developed a CASE tool that facilitates the use of the method, especially on complex cases.

Although many works have studied the combination of UML and HAZOP on computer systems, none we know of focuses on human-robot interactions. In this paper, we present the HAZOP method, the UML language and their combination in Section 2. In Section 3, we adapt HAZOP attributes for generic interpretations of UML use case and sequence diagrams. This approach is applied, in Section 4, to the analysis of a mobile manipulator robot developed in the PHRIENDS project [19] and to a robotic strolling assistant developed in the MIRAS project [17]. Section 6 presents a CASE tool to support our method and Section 7 concludes this paper.

Table 1: Generic HAZOP guidewords

Guideword	Interpretation
No/None	Complete negation of the design intention No part of the intention is achieved and nothing else happens
More	Quantitative increase
Less	Quantitative decrease
As Well As	All the design intention is achieved together with additions
Part of	Only some of the design intention is achieved
Reverse	The logical opposite of the design intention is achieved
Other than	Complete substitution, where no part of the original intention is achieved but something quite different happens
Early	Something happens earlier than expected relative to clock time
Late	Something happens later than expected relative to clock time
Before	Something happens before it is expected, relating to order or sequence
After	After Something happens after it is expected, relating to order or sequence

2. BACKGROUND

The HAZOP method was developed at the beginning of the seventies by ICI (Imperial Chemical Industries). In its original form, the HAZOP method was particularly adapted for the study of thermo-hydraulic systems. The objective of HAZOP analysis is twofold: identify hazards and propose recommendations aimed at reducing the associated risk. The HAZOP method is based on brainstorming done by a group of experts whose collective knowledge has sufficient coverage of the concerned system and application. Through the HAZOP method, a system is analyzed by holding a review of the systematic generation of deviations defined by the conjunction of parameters of the system (e.g., pressure, temperature...) and guidewords (e.g., no, more, less...) as presented in Table 1 (generic guideword list from the now obsolete Defence Standard 00-58 [5] and the IEC-61882 standard [12]). The HAZOP method has been adapted to different domains and can be found in many forms with a focus on process, on human error, on procedure, or on software. Modification of the method consists in adapting the list of parameters and the list of guidewords to the specific viewpoint. Even though the HAZOP method is efficient, the results may be questionable when the perimeter of the study is too vast (completeness problem) or when the guidewords are either too numerous or too limited for the analysis to be relevant. Another limitation is that there is no systematic method to adapt the guidewords to the considered domain, so adaptation depends on the expertise of the initiator(s) of the method. Additionally, the HAZOP method needs an appropriate allocation of human resources and suffers from combinatorial explosion when too many deviations are considered or when the practitioners go into too much detail.

Risk analysis is usually performed using a model of the system (e.g., a block diagram). With the advent of object-oriented languages and associated notations (such as UML), many studies have been carried out to determine how those new techniques could be used as input models for risk analysis techniques. UML is a standard general-purpose modeling language that includes a graphical notation enabling the representation of an abstract model of a system [18]. The UML model of a system is composed of different UML diagrams, each of which is a partial graphical representation of the system that concentrates on a particular viewpoint. Two diagrams are commonly used for description of the system usage: use cases and sequence diagrams. Use cases represent intended use of the system and are linked with the actors that can trigger scenarios of the use case. Each use case is further documented by fields such as pre and post conditions. Each sequence diagram represents one particular scenario of one use case.

Our risk analysis approach is based on a re-interpretation of the HAZOP guidewords presented in table 1 in the context of different UML models. The proposal in [16], followed by a more systematic study in [10], considers a guideword interpretation for the deviations of UML elements such as class, association, classifier role, message, etc. A similar approach was followed in [7] and [14], which also present a statistical analysis of the usability of this method. The guideword interpretation for the static UML diagrams in those studies aims to inspect the model to identify development faults rather than operational deviations. Nevertheless, for the UML dynamic diagrams (use case, sequence, activity, and statechart diagrams) many guideword interpretations can be used for exploring deviations during operational life. This is the case in studies presented in [15] and more formally in [2], which focus on use cases. The latter study led to a method that has been successfully used in [3] and [6]. This work on use cases also inspired a similar approach for security where new interpretations of guidewords have been proposed. Even if this work is more oriented towards malicious behavior of actors [21], several interpretations can be applied in safety-critical systems with human-machine interactions. In this paper, we build on the results of those studies, with a focus on use case and sequence diagrams in order to explore deviations during operational life. We also give a particular attention to the integration of HAZOP-like human error analysis techniques as presented in [8]. Indeed human factors methods [22] are a major issue in safety-critical systems but their analysis is often uncorrelated from preliminary system modeling activities. On the contrary, a key point of our approach is to consider human factors from the outset, by including them in the preliminary risk analysis.

3. UML-BASED HAZOP ANALYSIS

In this section, we present our method to analyse risks based on a UML description of human-robot interactions. The description is done using a subset of the UML use case and sequence diagrams. The risk analysis is then performed on this description using an adaptation of the HAZOP method.

3.1 UML interaction model

At first, the system goals must be represented using UML use cases. Use cases specify elementary objectives of use of the system (e.g., take an object from user hand). For each use case, a description is provided as well as conditions associated with it. A use case is described by:

- A name providing a unique identifier, for example “Call and autonomous movement of the robot”;

Table 2: Attributes, guidewords and interpretations for use case entity

Entity = Use Case		
Attribute	**Guideword**	**Interpretation**
Preconditions / Postconditions / Invariants	No/none	The condition is not evaluated and can have any value
	Other than	The condition is evaluated true whereas it is false The condition is evaluated false whereas it is true
	As well as	The condition is correctly evaluated but other unexpected conditions are true
	Part of	The condition is partially evaluated Some conditions are missing
	Early	The condition is evaluated earlier than required (other condition(s) should be tested before) The condition is evaluated earlier than required for correct synchronization with the environment
	Late	The condition is evaluated later than required (condition(s) depending on this one should have already been tested) The condition is evaluated later than required for correct synchronization with the environment

- An abstract describing the interaction that occurs in the main scenario of the use case, for example "When called by the user, the robot moves from its current position to a position near the user";

- A series of preconditions that must be satisfied before the use case can be executed, for example "The user called the robot" and "The robot is free from other tasks";

- A series of postconditions that must be satisfied after the use case has been completed successfully, for example "The robot is in the user's vicinity";

- A series of invariants that must be fulfilled throughout the execution of the use case, for example "The robot does not collide with the environment or the user".

UML sequence diagrams are then used to model the interactions between the robotic system and humans. Interaction between objects of the sequence diagram can be represented by messages while actions of one object can be represented using self-messages. We also use annotations to express the types of interaction (physical contact, visual signal, etc.) when the design is sufficiently advanced for that to be known.

For each use case, at least one sequence diagram should be drawn for the nominal scenario. Sequence diagrams should also be drawn for the most pertinent alternative scenarios. The exceptional scenarios can be ignored as they will be identified and analyzed during the HAZOP analysis.

This UML specification should be done as early as possible in the development process to allow early identification of major risks and consequent adaption of the design to meet the safety requirements of the robotic system. This is possible since the UML specification remains at a very high level of abstraction. The use case diagrams define the purpose of the system and the sequence diagrams of interest describe just the preliminary design of the system.

3.2 HAZOP method adaptation

Once the UML interaction model is completed, the HAZOP method is applied by selecting *elements* of a diagram and applying guidewords to them. In the Defence Standard 00-58 [5], the HAZOP analysis is the systematic identification of every deviation of every *attribute* of every *entity* (Figure 1). We define those terms as follows:

- An *entity* defines what part of the system model is under investigation. In our case it refers to a use case or a sequence diagram.

- An *attribute* refers to a physical or logical property of an *entity*:
  - For use cases, we choose the fields: (1) preconditions, (2) postconditions, and (3) invariants.
  - For sequence diagrams, we identify five attributes for each message: (1) predecessors and successors during the interaction, (2) message timing, (3) send and receive objects, (4) message guard condition and (5) message parameters.

Table 2 is the adaptation of the HAZOP guidewords for use cases, and Table 3 for sequence diagrams. An interpretation of the generic deviation is also provided in order to guide the mental process. These tables were derived from a combination of the different studies presented in section 2, discussions with experts, application of the guidewords to small case studies, models of computation errors, and confrontation with human error models.

Once deviations have been identified, possible consequences and causes are analyzed. To do this, the conditions of execution of the sequence diagram (e.g., environmental conditions or human states) need to be taken into account. The next step is to propose hints regarding possible risk reduction means to prevent the occurrence of deviations or to provide protection against their unwanted effects. One way of preventing the occurrence of deviations is to guarantee that a function or functional block whose failure can give rise to this deviation has a high level of integrity, i.e., it is sufficiently trustworthy to meet the safety objectives. For this, we use the concept of Safety Integrity Levels (SILs) as defined in the ISO/IEC61508 standard [11]. We consider that, for a safety-related function, the SIL is determined only in terms of the severity of the consequences of its failure. Hence, we used a direct mapping between severity levels and SILs. Of course, as presented in the standard, other approaches can be used to calculate a SIL. Such alternatives should be considered for each given project, depending on its safety objectives.

For some functions, it is difficult to meet the assigned SIL requirements. For example, the SIL assigned to a critical software component might require the use of stringent development methods and tools that are not capable of dealing with the complexity of the component. Moreover, some deviations just cannot be treated in this way. For example, the root cause of a human error cannot be mapped to a function to which a SIL can be assigned. For these reasons, other recommendations need to be given to limit the effects of the deviation, such as modifications of the specification, of system usage or of the human-machine interfaces.

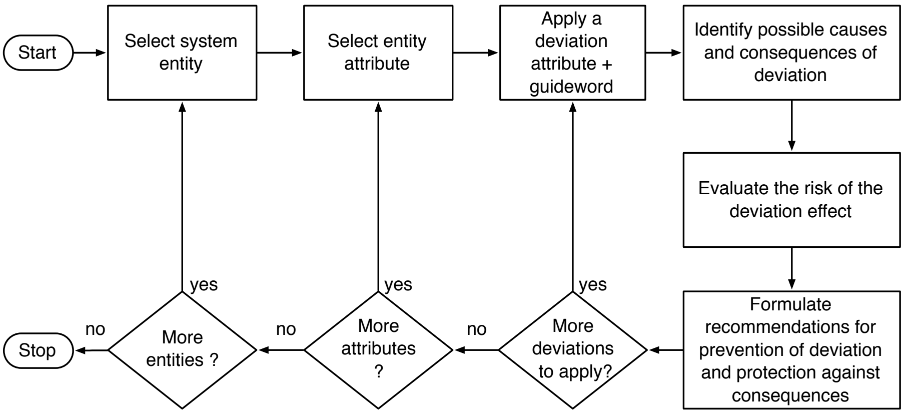

Figure 1: HAZOP methodology adapted from [5]

Table 3: Attributes, guidewords and interpretations for sequence diagram entity

Entity = Sequence Diagram		
Attribute	**Guideword**	**Interpretation**
Predecessors / successors during interaction	No	Message is not sent
	Other than	Unexpected message is sent
	As well as	Message is sent as well as another message
	More than	Message sent more often than intended
	Less than	Message sent less often than intended
	Before	Message sent before intended
	After	Message sent after intended
	Part of	Only a part of a set of messages is sent
	Reverse	Reverse order of expected messages
Message timing	As well as	Message sent at correct time and also at incorrect time
	Early	Message sent earlier than intended time
	Later	Message sent later than intended time
Sender / receiver objects	No	Message sent to but never received by intended object
	Other than	Message sent to wrong object
	As well as	Message sent to correct object and also an incorrect object
	Reverse	Source and destination objects are reversed
	More	Message sent to more objects than intended
	Less	Message sent to fewer objects than intended
Message guard condition	No/none	The condition is not evaluated and can have any value (omission)
	Other than	The condition is evaluated true whereas it is false, or vice versa (commission)
	As well as	The condition is well evaluated but other unexpected conditions are true
	Part of	Only a part of condition is correctly evaluated
	Late	The condition is evaluated later than required (other dependent condition(s) have been tested before)
		The condition is evaluated later than correct synchronization with the environment
Message parameters / return parameters	No/None	Expected parameters are never set / returned
	More	Parameters values are higher than intended
	Less	Parameters values are lower than intended
	As Well As	Parameters are also transmitted with unexpected ones
	Part of	Only some parameters are transmitted
		Some parameters are missing
	Other than	Parameter type / number are different from those expected by the receiver

The final outcome of a UML-HAZOP analysis consists of a list of recommendations and a list of hazards, together with the possible deviations leading to them. This list of hazards may be converted to a list of risks when the probabilities of occurrence of the deviations can be estimated (a risk is a combination of a harm probability and severity [13]). This is possible when the design is sufficiently well advanced to allow the use of other risk analysis methods such as Fault Tree Analysis.

3.3 HAZOP analysis table

To assist the HAZOP process, we propose a deviation analysis table with the following columns (cf. example given Table 5[1]):

1. Element: the UML element on which the deviation is applied.

2. Attribute: the considered attribute.

[1]For compactness, items 1 and 2 and items 6 and 7 are grouped into single columns in this table.

3. Guideword: the applied guideword.

4. Deviation: the deviation resulting from the combination of the attribute and the guideword.

6. Use Case Effect: effect at the use case level.

7. Real World Effect: possible effect in the real world.

8. Severity: rating of effect of the worst case scenario in the real world.

9. Possible Causes: possible causes of the deviation (software, hardware, human, etc.).

10 Integrity Level Requirements: a preliminary safety integrity level [11] aimed at avoiding the deviation with a sufficient level of confidence (this will lead to the application of specific fault prevention and fault removal techniques [4]).

11. New Safety Requirements: if the deviation cannot be avoided, new requirements are specified (e.g., additional fault tolerance techniques, or regulatory constraints).

12. Remarks: explanation of analysis, additional recommendations, etc.

13. Hazard Numbers: real world effects are identified as hazards and assigned a number, helping the users to navigate between results of the study and the HAZOP tables.

4. CASE STUDIES

This section presents two applications of the method. The first study was performed for the PHRIENDS project [19]. It analyzed the safety of a robotic mobile manipulator. The second study was carried out in the framework of the MIRAS project [17] and analyzed safety of a robotic strolling assistant. For both studies, we rated the severity of deviations (column 8 of the HAZOP analysis table) according to the abbreviated injury scale of [1].

4.1 Application to a mobile robot manipulator

Figure 2: Example of a mobile manipulator: "concept omniRob" © at Automatica 2008 exhibition – KUKA Roboter GmbH

Table 4: Description of UC4 "Take an object from the user's hand"

Use case name	**UC4. Take an object from the user's hand**
Abstract	The user orders the robot to take an object from his hand
Precondition	No object in the gripper Location reachable Object can be taken
Postcondition	Robot base is stopped Object in the gripper Robot arm is in transportation position
Invariant	None

The first considered system is a wheel-based mobile robot with a manipulator arm (Figure 2). The environment is a workshop and factory with human workers. Collaborative work between a human and the robot is possible (e.g., the robot can give an object to the human). The robot is able to navigate in a dynamic environment where there are other mobile objects (e.g., humans). Identified use cases are: *Take an object from a specified location* (UC1), *Place an object at a specified location* (UC2), *Go to a location (holding or not holding an object)* (UC3), *Take an object from the user's hand* (UC4), *Give an object to the user* (UC5), *Abort a task* (UC6), *Guide the robot arm to a location* (UC7), *Pause and resume a task* (UC8), and *Physical interaction with the arm* (UC9).

The five first use cases do not necessarily imply physical contact or even an interaction via an object; they can nonetheless be interrupted by physically stopping the arm of the robot (UC9) in order to switch to one of the use cases UC6, UC7 or UC8. Two more use cases are *Program robot* (UC10) and *Set up* (UC11), which, although they can induce major safety problems, have not been considered here since they are quite common use cases in industrial robotics and do not introduce any novelties with respect to human-robot interaction.

For each use case, preconditions, postconditions and invariants were identified, and the nominal scenario was modeled using a sequence diagram. By way of an example, Table 4 shows the description of UC4 *Take an object from the user's hand* and Figure 3 presents the sequence diagram of the nominal scenario of UC4. Table 5 presents an extract of the study of this sequence diagram. Analysis of the first deviation in this table leads to the requirement of a protocol for communication between user and robot. Analysis of the second deviation in Table 5 leads to the identification of a safer human-robot interaction for passing an object (Remarks column). It is suggested that the robot's behaviour has to be modified.

During this study, 130 elements were analyzed leading to 1694 deviations. However, only 768 deviations (45%) could be interpreted. The sample list of hazards presented in Table 6 is extracted from the full set of HAZOP tables in which 21 hazards were identified. Due to space limitations the table shown does not contain the extra column with the list of sources of each hazard class (this column is contained in our study and in the tool presented in Section 6). This haz-

Table 5: Extract of SD4 “Take an object from the user’s hand” HAZOP analysis

Project : PHRIENDS — **Date: June-01-2008**
HAZOP number : UC4/SD4 — **Prepared by: Ofaina Taofifenua**
Entity : Sequence Diagram 4 (sd4) “Take an object from the user’s hand” — **Revised by: Jérémie Guiochet**
Approved by:

Element (attribute)	Guide word	Deviation	a. Use Case Effect b. Real World Effect	Severity	Possible Causes	Integrity level Requirements	New Safety Requirements	Remarks	Hazard Number
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
Receive and interpret order (pred/succ)	More than / as well as	The robot receives several different orders	a. Wrong order taken into account b. Wrong task, bad synchro-nization between robot and user, could result in collision	**Moderate**	Failure of H/W for order reception Human error	H/W for order reception should be SIL1	User education and training Define a protocol for communication between user and robot (e.g. acknowledgment messages, user can check interpretation of the order)	Means for communication between robot and user needs to be defined for the PHRIENDS use case (speech, graphical HMI, vision, etc.)	
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
Put the object in the gripper (pred/succ)	Before	Since the gripper is open the user can give the object to the robot before the latter is ready	a. Bad synchronization between user and robot can cause collision b. The object can fall / The arm and human can collide	**Severe**	Human error	None	The robot should keep the gripper closed until the arm movement is finished	The procedure in the seq. diag. is as follows: the robot opens its gripper then the robot arm moves towards the user hand. Only then the user can place the object in the robot gripper. A safer procedure is: the robot should keep the gripper closed until arm movement is finished -> modify sequence diagram	2, 19, 20
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	

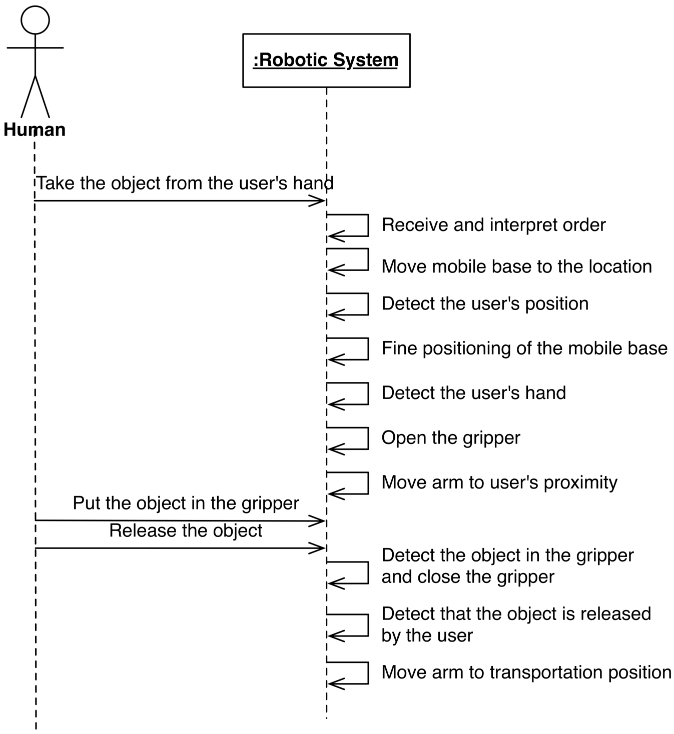

Figure 3: Sequence diagram SD4 giving main scenario of UC4 “Take an object from the user’s hand”

ard list was checked by robotics experts of the PHRIENDS project (KUKA Roboter GmbH). The analysis led to 18 high-level recommendations, for example:

- R1. The user must be able to stop the robot at any time by touching any part of the robot.

Table 6: Extract of identified hazards (total number 21)

Hazard	Hazard description
1	Robot base is moving while it should not
2	Robot arm is moving while it should not
20	Task planning error (fault in the planner or insufficient knowledge of the environment or of the nature of the object)
21	Gripper speed is too slow for human/robot synchronization

- R4. The robot and the user have to be aware of each other: some device or means should be used to communicate to the user the actual mode of operation of the robot.

- R6. Allow the user to guide not only the robot arm but the mobile base too.

4.2 Application to a robotic strolling assistant

The second considered system is a robotic strolling system that helps partially-disabled persons to stand up, stroll and sit down. It is intended to be used in elderly care centers by people suffering from gait and orientation problems. The system consists of a wheeled base and a moving handlebar (cf. Figure 4), and is equipped with several sensors to detect physiological parameters and the posture of the patient. It can also move autonomously. The preliminary design of the robot identified 11 use cases: *Strolling* (UC01), *Standing up operation* (UC02), *Sitting down operation* (UC03), *Balance loss handling* (UC04), *Call and autonomous movement of the robot* (UC05), *End of use detection and movement to a waiting position* (UC06), *Positioning the robot by hand* (UC07), *Alarms handling* (UC08), *Patient profile programming* (UC09), *Patient profile learning* (UC10), and *Robot set-up* (UC11).

Figure 4: Robuwalker – First prototype

Table 7: UC02 "Standing up operation"

Use case name	UC02. Standing up operation
Abstract	The patient stands up with help from the robot
Precondition	The patient is sitting down The robot is waiting for the standing up operation Battery charge is sufficient to do this task and to help the patient to sit down again The robot is in front of the patient
Postcondition	The patient is standing The robot is in admittance mode
Invariant	The patient holds both handles of the robot The robot is in standing up mode Physiological parameters are acceptable

When the risk analysis was carried out, the design of this system was in an earlier stage than for the robot manipulator. Especially UC09, 10 and 11 were not specified at the time the analysis was performed. Table 7 presents the conditions linked to the UC02, *Standing up operation*. The nominal scenario of this use case is shown in Figure 5.

Out of 993 generated deviations, 297 (30%) were analyzed and 157 led to the identification of 13 main hazards (the other deviations had minor effects). An extract of the hazard list can be found in Table 8. Following the analysis, 26 high-level recommendations and 17 new safety requirements were issued, for example:

- Filter patient force to avoid oscillation amplification by the robot,
- Send regularly a network heartbeat from the robot.

Table 8: Extract of identified hazards (total number 13)

Hazard Number	Hazard description	Occurrences
1	Incorrect patient position during robot use	7
2	Fall of the patient during robot use	28
12	Imbalance of the patient caused by the robot	33
13	Patient tiredness	28

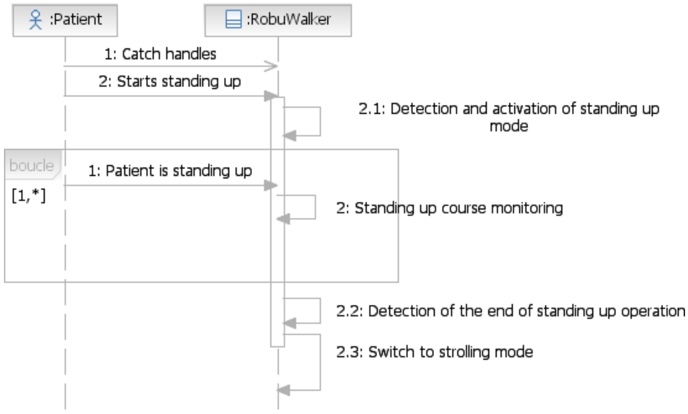

Figure 5: Sequence diagram UC02.SD01 giving main scenario of UC02 "Standing up operation"

Launch alarm on time-out,

- Worst-case electrical consumption must be evaluated beforehand.

The application of the UML-HAZOP approach has been compared to a preliminary hazard analysis (PHA [20]) carried out at the beginning of the project during two workshops with robotic experts. An important result is that our approach identifies all human-robot interaction hazards already identified by the PHA but also new hazards (e.g., a situation where the user is isolated and the system does not have enough power to call the medical staff). Another important result is that all recommendations were approved by robotic experts in the MIRAS project (ISIR[2]). The recommendations were labeled according to the different versions of the prototype (development, validation and final). The second robot prototype will include the corresponding recommendations given by this analysis.

5. QUALITY OF THE METHOD

To assess the quality of our approach, we analyzed it from four different perspectives: a) **integrability**, how well does it integrate with the development process? b) **usability**, is it easy to use? c) **validity**, are the results complete? d) **applicability**, can the results be used?

Integrability: the method was designed to be used at the early development stage and can be refined during the design process. Furthermore, it uses common UML for modeling the system and can thus be integrated in a normal development process. In the MIRAS project, all the UML models have been shared and co-designed with the development team.

Usability: Table 9 shows statistics resulting from application of the method to each study. It can be seen that many more deviations were analyzed in the PHRIENDS study than in the MIRAS one. This is mainly because the MIRAS project is still ongoing so its design is less detailed. The combinatorial aspect of the method, which is a common

[2]Institut des Systèmes Intelligents et de Robotiques, Paris, France

Table 9: Application of the method – Statistics

Project	PHRIENDS	MIRAS
Use Cases	9	11
Conditions	39	45
Analyzed deviations	297	317
Interpreted deviations	179 (60.3%)	134 (42.3%)
Interpreted deviations with recommendation	120 (40.4%)	72 (22.7%)
Sequence diagrams	9	12
Messages	91	52
Analyzed deviations	1397	676
Interpreted deviations	589 (42.2%)	163 (24.1%)
Interpreted deviations with recommendation	274 (19.6%)	85 (12.6%)
Totals:		
UML Elements	130	97
Analyzed deviations	1694	993
Interpreted deviations	768 (45.33%)	297 (29.9%)
Interpreted deviations with recommendation	394 (23.25%)	157 (15.8%)

drawback when using HAZOP, was manageable using classical Excel spreadsheets. However, we believe an appropriate tool would be of assistance in this respect.

Another important point defining the usability of the method is that it is easy to understand by non-experts thanks to the UML model and the HAZOP method. Indeed, UML is really common and little expertise is needed to understand the chosen subset of UML. The HAZOP methodology is simple and we have successfully presented it within an hour to our project partners.

Flexibility to design modifications is another important point of the method's usability. When diagrams change, the deviations corresponding to new elements must be created and deviations corresponding to removed elements must be deleted. When physical changes are made on the system (e.g. bumpers around the system to reduce the impact of a collision), hazard numbers enable the corresponding deviations to be found in order to modify the HAZOP tables. Those two points make the method fairly flexible to design changes. However, applying those changes can be time-consuming when using a standard UML tool and spreadsheet software. Again, an appropriate tool would be useful from this viewpoint.

Validity: as previously mentioned, a Preliminary Hazard Analysis (PHA) had been carried out in the MIRAS project before applying our UML-HAZOP approach. More operational hazards were identified by our method than by the PHA. Note that the common hazards coming from the use of electric machines (electrocution, mechanical projections, etc.) are not covered by our approach, so a complete safety analysis should also integrate methods such as PHA.

Finally, we can draw a positive conclusion about our choice of guidewords. In the first study, all selected guidewords except two ("less than" and "part of" of the predecessor/successor attribute) led to interpreted deviations. In the second study, all selected guidewords except one ("reverse" of the predecessor/successor attribute) led to interpreted deviations (and to recommendations). Neither study used sender/receiver attribute guidewords other than "No" because both systems only considered one human and one robot. However, when modelling systems designed to work with several humans or multiple robots, those guidewords be used. Another interesting point is that the two studies were carried out by two different analysts. This explains the use of slightly different guideword depending on the analyst. However, the guidewords lists appear to be complete enough to be used by different analysts with different interpretations.

Applicability: the analyses generate several artifacts: hazard list, HAZOP tables, recommendation list and integrity requirement list. The hazard list enables the identification of major risks of the system. It is linked to a series of safety recommendations to reduce the occurrence or the severity of hazards. In both studies, the hazards and recommendations were accepted by the robotics partners and integrated into the development process. The integrity requirement list leads to significant recommendations from the IEC-61508 standard [11] that are readily exploitable. For certification, the various artifacts can be provided as documention for the measures taken to ensure the safety of the system. Moreover, they are quite concise: we were able to present them to our project partners in a couple of hours.

We therefore conclude positively about the method: it can be easily integrated into a normal development process, it covers the major operational hazards within its scope, and leads to significant recommendations. Although it is usable using standard tools, we decided that a specific tool would be better to handle complex cases and design modifications.

6. TOOL DESCRIPTION

To ease the analysis of complex systems, we developed a CASE tool to support the method. It helps to manage the combinatorial aspects of the HAZOP method by maintaining consistency between UML models and HAZOP tables and by providing document generation and management features. The tool is built as an Eclipse plugin (www.eclipse.org) using the Graphical Modelling Framework (GMF). In this tool (Figure 6), the analyst can draw UML use-case and sequence diagrams. Using guideword templates, HAZOP tables are automatically generated, ready to be filled out by the analyst.

The analyst can first model the system using use-case diagrams created via a drawing view (view 3 of Figure 6). The toolbox (view 4) enables various elements to be added to the diagram and a property view enables use case conditions to be entered. View 2 shows the diagram view of a sequence diagram. Once the system is modeled, the HAZOP table can be edited using the HazopTable view (view 7). The list of guidewords, the list of columns and the list of severities are editable using the main project view (view 1). Using this template, the list of deviations is automatically generated (view 7). The analyst can then select applicable deviations and fill the corresponding columns. Fast search of specific deviations is available through field 5. Also, when selecting a UML element in a diagram, the corresponding deviations are automatically shown. When filling the table, the recommendation list and corresponding hazards are automatically generated in the project view. The toolbox of the HazopTable view (6) enables deviations to be added (for example,

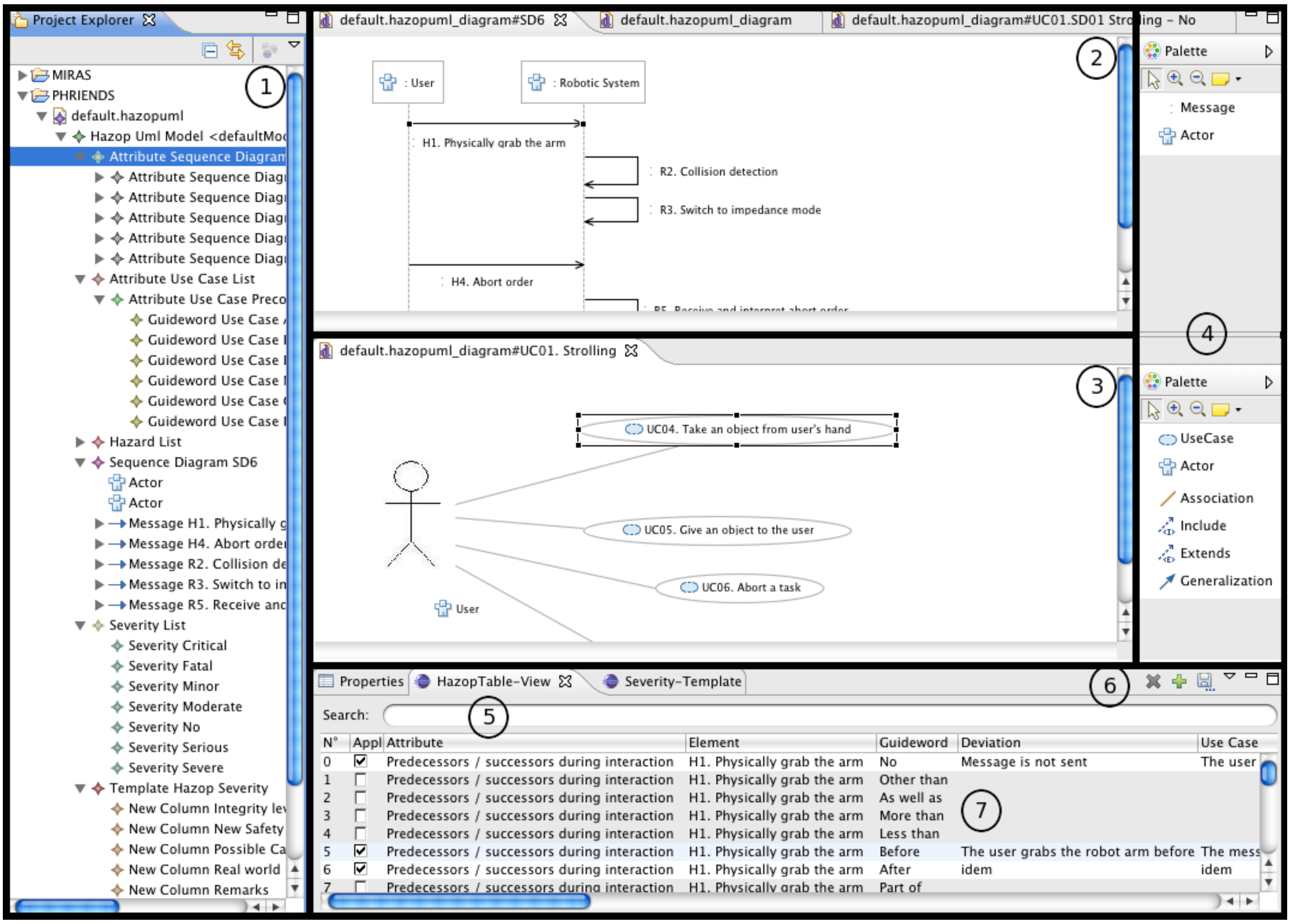

Figure 6: Main view of the CASE Tool to support the UML-HAZOP method

several deviations for the same keyword) and to export the current table in the CSV (Comma Separated Values) format readable by spreadsheet software. Diagrams can be exported in the image format. A report generator is currently under development.

The tool is easy to use because of its simplicity and integration to a common environment. It manages the combinatorial aspects of the HAZOP method by automatic generation of partially filled-out deviations. The method and guideword list can be adapted thanks to HAZOP table templates. Furthermore, the analysis can be exported in CSV to reuse the results outside the tool. However, rich formats like HTML or Excel are not yet available for exportation, which currently limits the integration of our tool with other software. Rich format exportation should permit the generation of the artifacts identified in our case studies: use case list with conditions, sequence diagrams, list of remarks issued during the analysis (incomplete specification, useful relation to existing norms, etc.), list of generated hazards, HAZOP tables, list of recommendations, and list of integrity level requirements. Since we only use a partial subset of UML, the tool cannot be used for the whole modeling process. However, importation and exportation to other software like IBM Rational Software Architect is a planned feature.

7. CONCLUSION

To tackle safety of robotic systems, appropriate analysis methods are needed. Classic methods suffer from several limitations: unsuited for human-robot interaction, inability to cope with multiple stakeholders and too late implication in the development process. We proposed an adaptation of the HAZOP method to apply it on a subset of the Unified Modeling Language. The method is particularly aimed at modeling physical human-robot interaction early in the development process. The discussions between stakeholders are facilitated through the use of a well-known standard format (UML). Furthermore, since the process is quite systematic, very few analysts are needed once the system is modeled. The combinatorial aspect of the HAZOP method remains manageable since the analysis is restricted to the use case diagram and context sequence diagrams (showing only actors and the system). The developed tool also helps considerably in this respect since it facilitates navigation between generated summary listing and rough analysis contained in the HAZOP tables.

The method has been applied to two systems: a robotic mobile manipulator and a robotic strolling assistant. It led to the identification of, respectively, 18 and 26 recommendations to increase the safety of those systems. The recommendations were accepted and taken into account by our partners in both projects. Thus, we believe the method is usable and leads to significant recommendations.

To ease the application of our method, a CASE tool was developed to partially automate the generation of deviations and to manage necessary book-keeping. Further developments are planned to finalize the tool, especially rich format exportations/importations and user interface improvements.

We plan to improve the method further by specializing guidewords for different kinds of message (self-message or interaction) and for different kinds of conditions (precondition, postcondition or invariant). With this specialization, we should reduce the number of proposed deviations, keeping only significant ones.

8. ACKNOWLEDGEMENT

This work was partially supported by the PHRIENDS Specific Targeted Research Project, funded under the 6th Framework Programme of the European Community under Contract IST-045359. It was also partially supported by the MIRAS Research Project, funded under the TecSan (Technologies for Healthcare) programme of the French National Research Agency (ANR).

9. REFERENCES

[1] AIS98. The abbreviated injury scale. Technical report, Association for the Advancement of Automotive Medicine, Des Plaines, IL, USA, 1998.

[2] K. Allenby and T. Kelly. Deriving safety requirements using scenarios. In *Requirements Engineering, 2001. Proceedings. Fifth IEEE International Symposium on*, pages 228–235, 2001.

[3] A. Arlow, C. Duffy, and J. McDermid. Safety specification of the active traffic management control system for english motorways. In *The First Institution of Engineering and Technology International Conference on System Safety*, 2006.

[4] A. Avizienis, J.-C. Laprie, B. Randell, and C. Landwehr. Basic Concepts and Taxonomy of Dependable and Secure Computing. *IEEE Transactions on Dependable and Secure Computing*, 1(1):11–33, Jan. 2004.

[5] DefStan00-58. HAZOP studies on systems containing programmable electronics. Defence Standard, Ministry of Defence, UK, 2000.

[6] I. Frantz, G. Andy, M. John, and I. Toyn. Integrating safety and formal analyses using UML and PFS. *Reliability Engineering and System Safety*, 92(2):156–170, 2007.

[7] J. Gorski and A. Jarzebowicz. Development and validation of a HAZOP-based inspection of UML models,. In *3rd World Congress for Software Quality, Munich, Germany*, 2005.

[8] J. Guiochet, G. Motet, C. Baron, and G. Boy. Toward a human-centered UML for risk analysis - application to a medical robot. In C. Johnson and P. Palanque, editors, *Proc. of the 18th IFIP World Computer Congress (WCC), Human Error, Safety and Systems Development (HESSD04)*, pages 177–191. Kluwer Academic Publisher, 2004.

[9] J. Guiochet and A. Vilchis. Safety analysis of a medical robot for tele-echography. In *Proc. of the 2^{nd} IARP IEEE/RAS joint workshop on Technical Challenge for Dependable Robots in Human Environments, Toulouse, France*, pages 217–227, 2002.

[10] K. M. Hansen, L. Wells, and T. Maier. HAZOP analysis of UML-based software architecture descriptions of safety-critical systems. In *Proceedings of NWUML*, 2004.

[11] IEC61508. Functional safety of electrical/electronic/programmable electronic safety-related systems. International Electrotechnical Commission, 2000.

[12] IEC61882. Hazard and operability studies (HAZOP studies) – Application guide. International Electrotechnical Commission, 2001.

[13] ISO/IEC-Guide51. Safety aspects - Guidelines for their inclusion in standards. International Organization for Standardization, 1999.

[14] A. Jarzebowicz and J. Górski. Empirical evaluation of reading techniques for UML models inspection. *ITSSA*, 1(2):103–110, 2006.

[15] P. Johannessen, C. Grante, A. Alminger, U. Eklund, and J. Torin. Hazard analysis in object oriented design of dependable systems. In *2001 International Conference on Dependable Systems and Networks, Göteborg, Sweden*, pages 507–512, 2001.

[16] K. Lano, D. Clark, and K. Androutsopoulos. Safety and security analysis of object-oriented models. In *SAFECOMP '02: Proceedings of the 21st International Conference on Computer Safety, Reliability and Security*, pages 82–93, London, UK, 2002. Springer-Verlag.

[17] MIRAS. Multimodal Interactive Robot for Assistance in Strolling. Project supported by the French ANR (National Research Agency) under the TecSan (Healthcare Technologies) Program (ANR-08-TECS-009-04), http://www.miraswalker.com/index.php/en.

[18] OMG-UML2. OMG unified modeling language (OMG UML), superstructure, v2.1.2. Object Management Group, formal/2007-11-02, 2007.

[19] PHRIENDS. Physical Human-Robot Interaction: Dependability and Safety. Project supported by the European Commission under the 6th Framework Programme (STReP IST-045359), http://www.phriends.eu/.

[20] M. Rausand and A. Høyland. *System Reliability Theory: Models, Statistical Methods and Applications, 2nd Edition.* Wiley, 2004.

[21] T. Srivatanakul. *Security Analysis with Deviational Techniques.* PhD thesis, University of York, 2005.

[22] N. Stanton, P. Salmon, G. Walker, C. Baber, and D. P. Jenkins. *Human Factors Methods: A Practical Guide for Engineering And Design.* Ashgate Publishing, 2006.