Université de Toulouse

LAAS-CNRS

IDENTIFICATION DE PROPRIÉTÉS DE SÛRETÉ VÉRIFIABLES À L'EXÉCUTION POUR LES SYSTÈMES AUTONOMES

Jérémie GUIOCHET

Jeremie.guiochet@laas.fr

Quynh Anh DO HUONG, Mohamed KAANICHE,

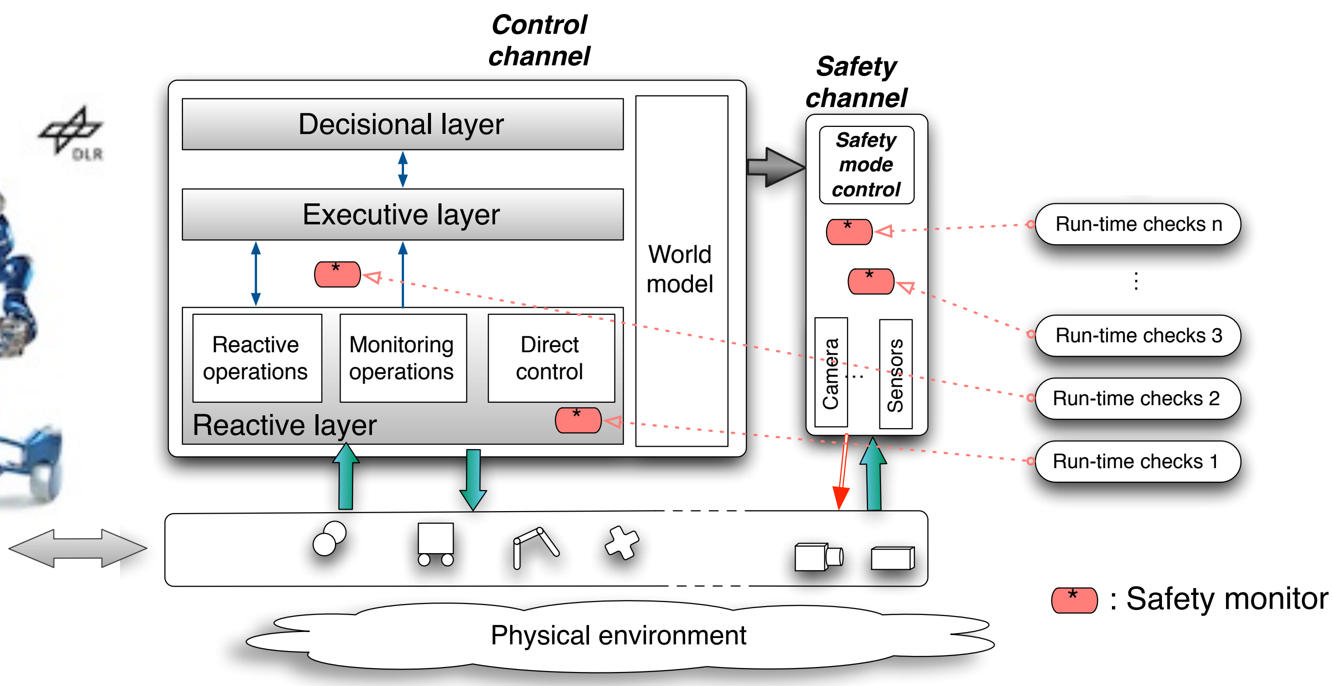
Amina Mekki-MOKTHAR, David POWELL, Mathieu ROY

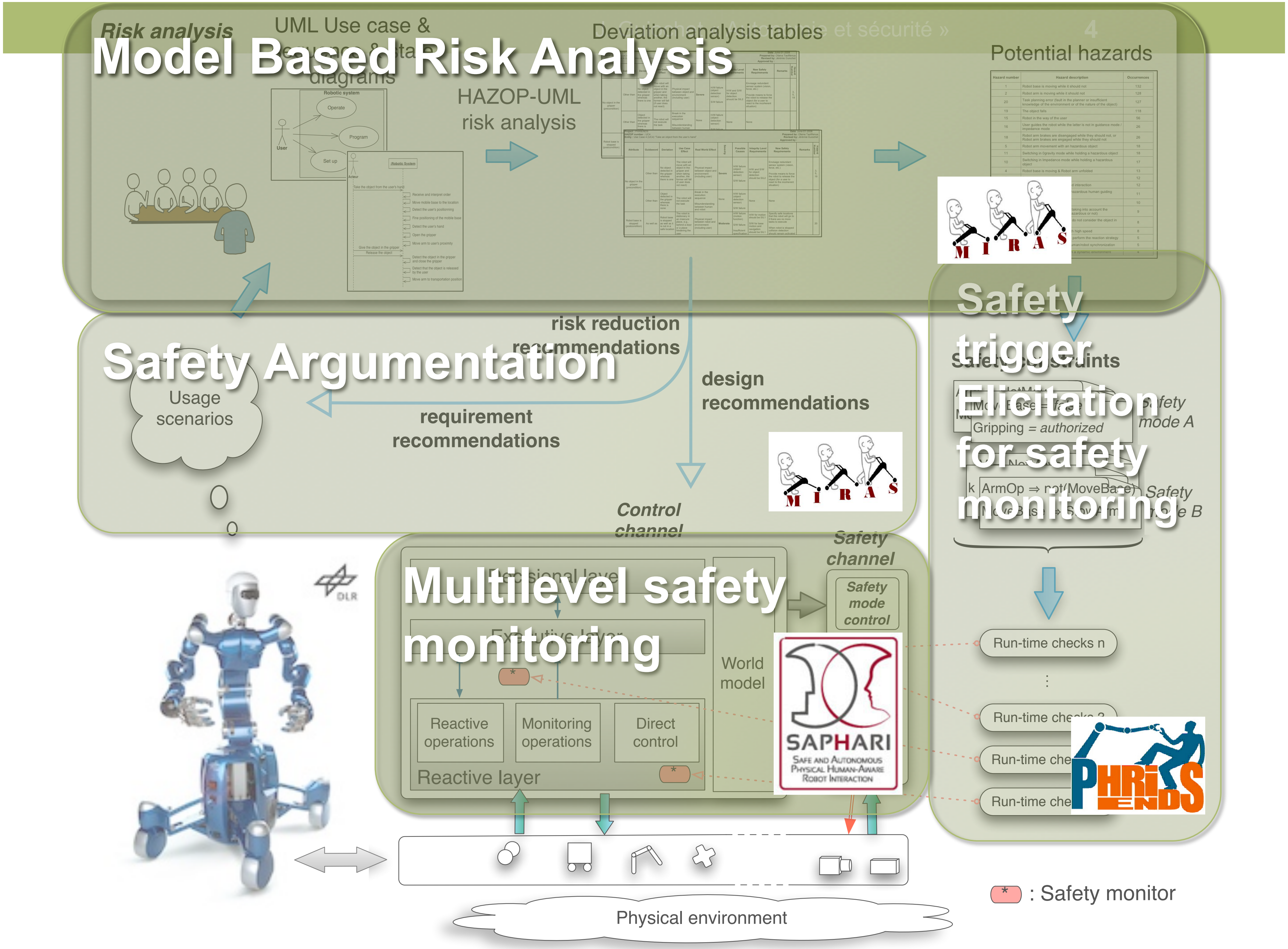
Atelier « Autonomie des robots et sécurité », GDR Robotique, Onera, Toulouse, 4 Juin 2012

Autonomous Systems & Safety

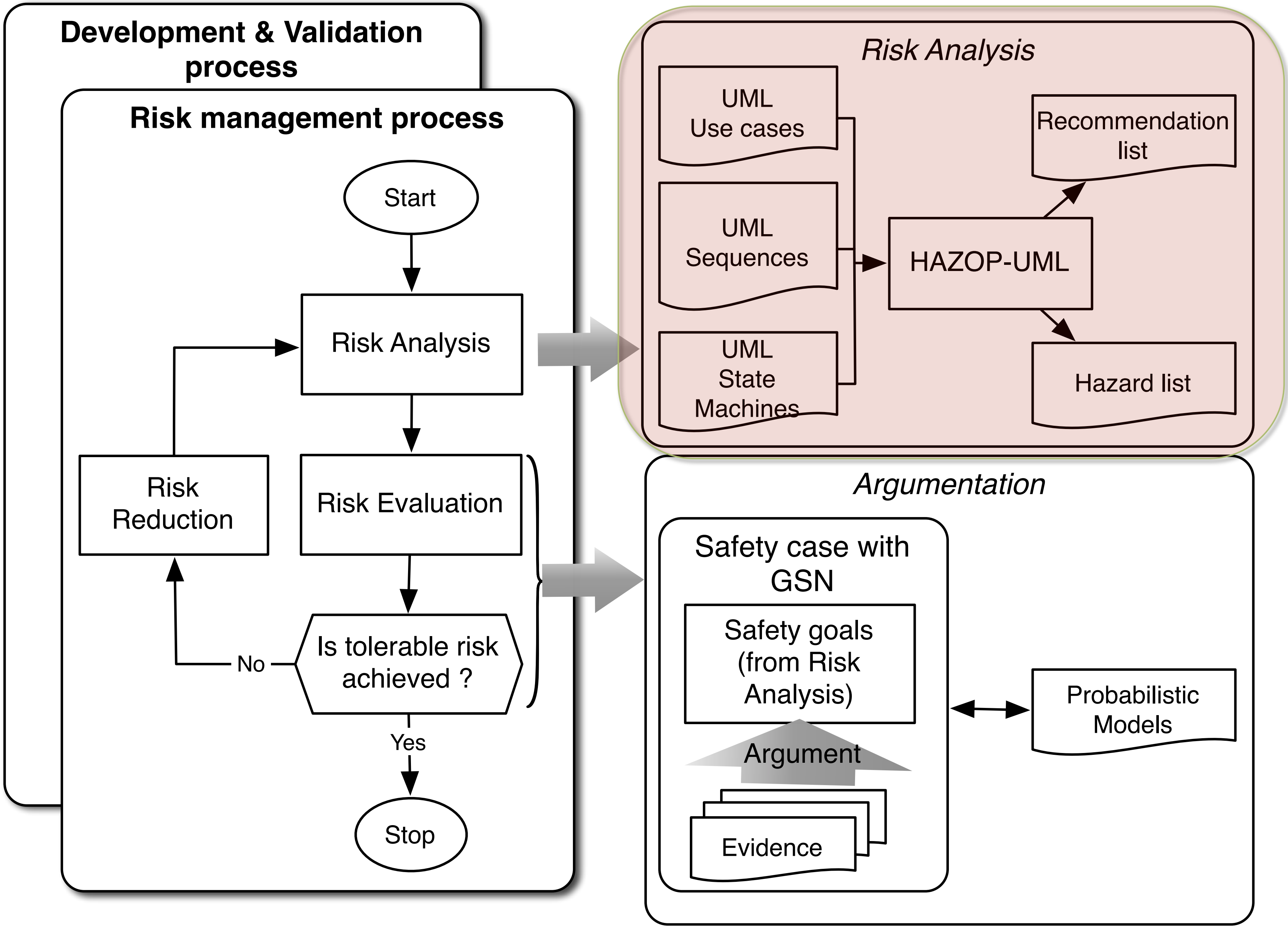
- Complex
  - architectures (e.g., different levels of abstraction)
  - interactions (e.g., humans, others systems)
  - technologies (e.g., HW/SW for perception)
- Moving in non structured environment
  - non deterministic behaviour -> non reproducible
  - uncertainties for environment perception

SAFE BY DESIGN ? / SAFETY ARGUMENTATION ?


Control channel
Safety channel
Decisional layer
Executive layer
World model
Safety mode control
Reactive operations
Monitoring operations
Direct control
Reactive layer
Camera
Sensors
Run-time checks n
Run-time checks 3
Run-time checks 2
Run-time checks 1
Physical environment
: Safety monitor

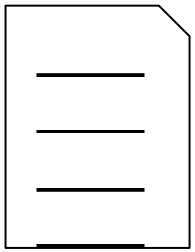


Model Based Risk Analysis
Risk analysis
UML Use case & sequence diagrams
HAZOP-UML risk analysis
Deviation analysis tables
Potential hazards
Safety Argumentation
risk reduction recommendations
design recommendations
requirement recommendations
Usage scenarios
Safety trigger Elicitation for safety monitoring
Safety constraints
Gripping = authorized
Safety mode A
ArmOp ⇒ not(MoveBase)
Safety mode B
Multilevel safety monitoring
Control channel
Safety channel
Safety mode control
World model
Reactive operations
Monitoring operations
Direct control
Reactive layer
Run-time checks n
SAPHARI
Safe and Autonomous Physical Human-Aware Robot Interaction
Physical environment
: Safety monitor

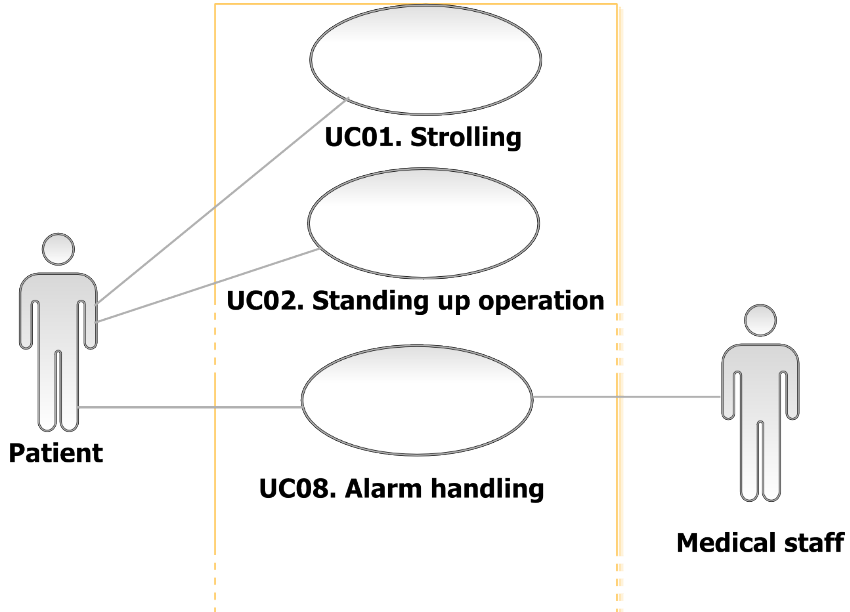

Development & Validation process
Risk management process
Start
Risk Analysis
Risk Evaluation
Risk Reduction
Is tolerable risk achieved ?
No
Yes
Stop
Risk Analysis
UML Use cases
UML Sequences
UML State Machines
HAZOP-UML
Recommendation list
Hazard list
Argumentation
Safety case with GSN
Safety goals (from Risk Analysis)
Argument
Evidence
Probabilistic Models

Unified Modeling Language

- Use cases
  - Describe the intended use of the robot
  - Completed with conditions

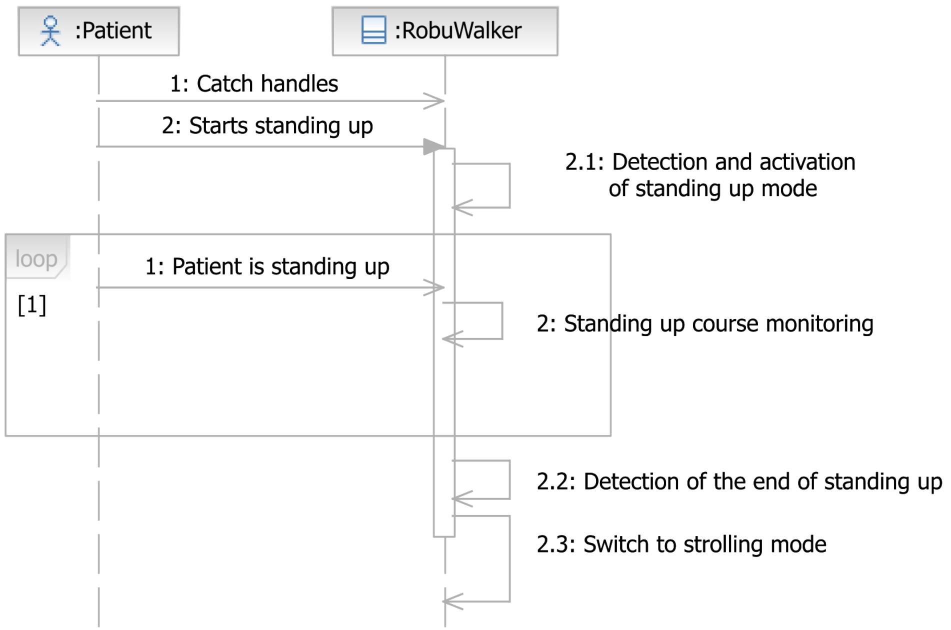


Textual description of:
- Preconditions
- Postconditions
- Invariants



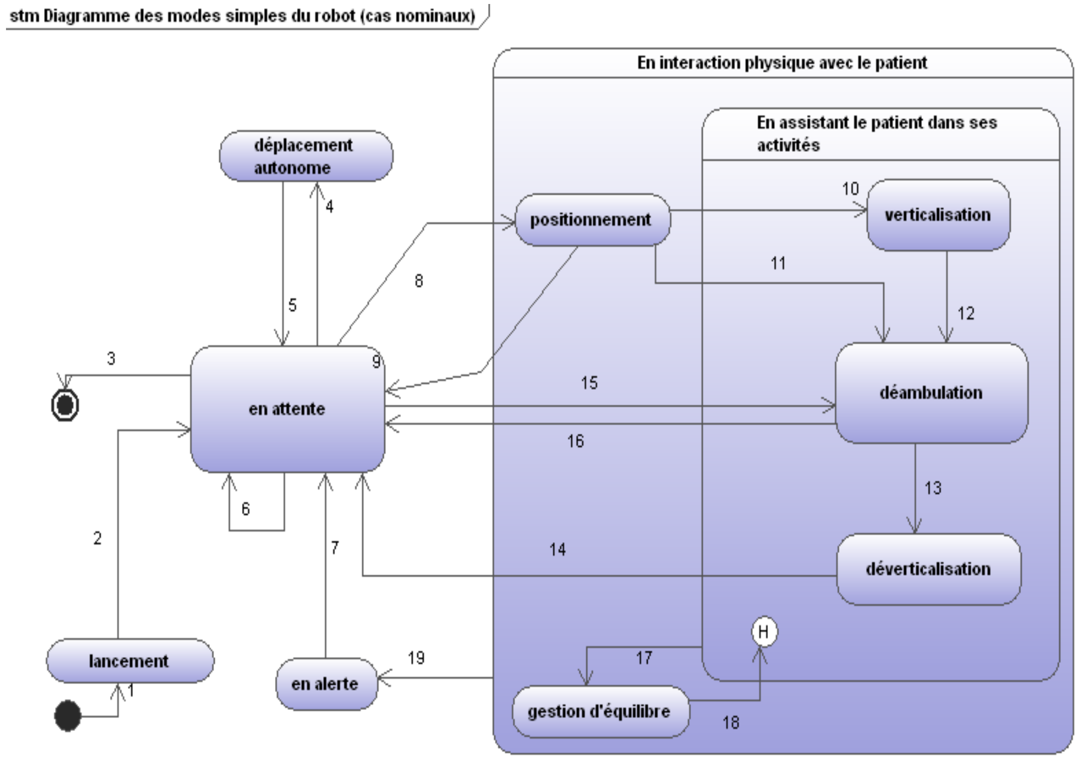
Unified Modeling Language

- Sequence diagrams
  - Describe nominal scenarios corresponding to the use cases
  - Messages are either actions (self-messages) or interactions

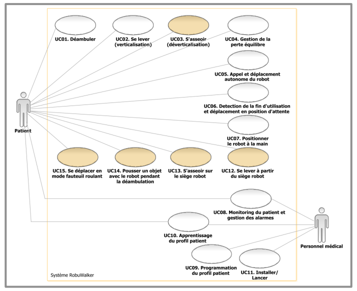


Unified Modeling Language

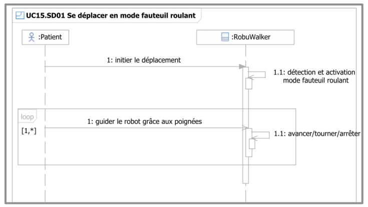
- Statechart
  - Describe different system's state
  - Completed with conditions



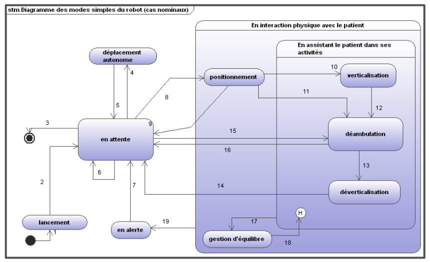
UML Models



Use Case Diagram



Sequence Diagram



Statechart

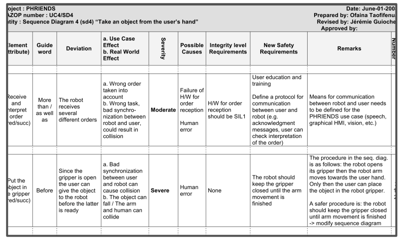
×

HAZOP Guidewords

Guideword	Signification
No / None	Complete negation of the design
More than	Quantitative increase
Less than	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

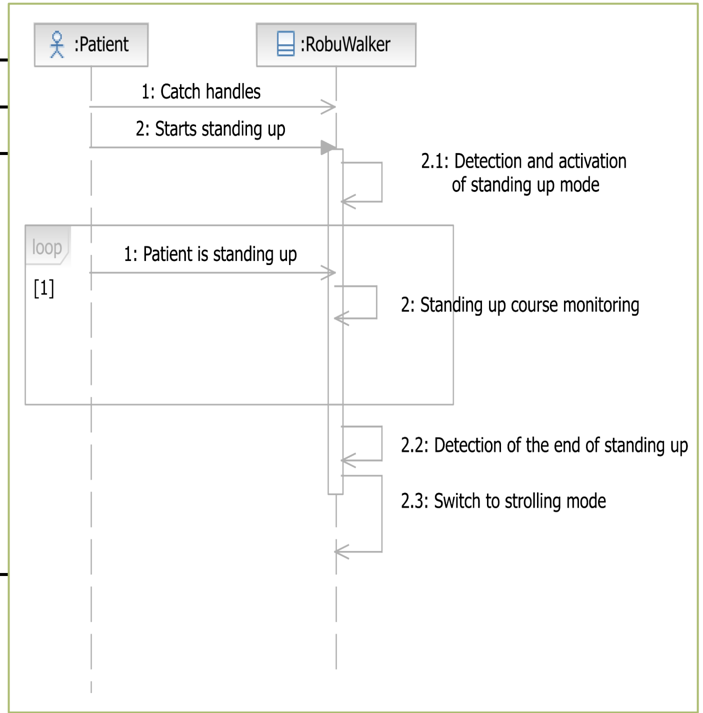
=

Risk analysis HAZOP-UML



HAZOP-UML

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 a s	Message sent at correct time and also at incorrect tim e
	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 objec t
	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



Example of HAZOP-UML application

Project : PHRIENDS
HAZOP number : UC4/SD4
Entity : Sequence Diagram 4 (sd4) “Take an object from the user’s hand”

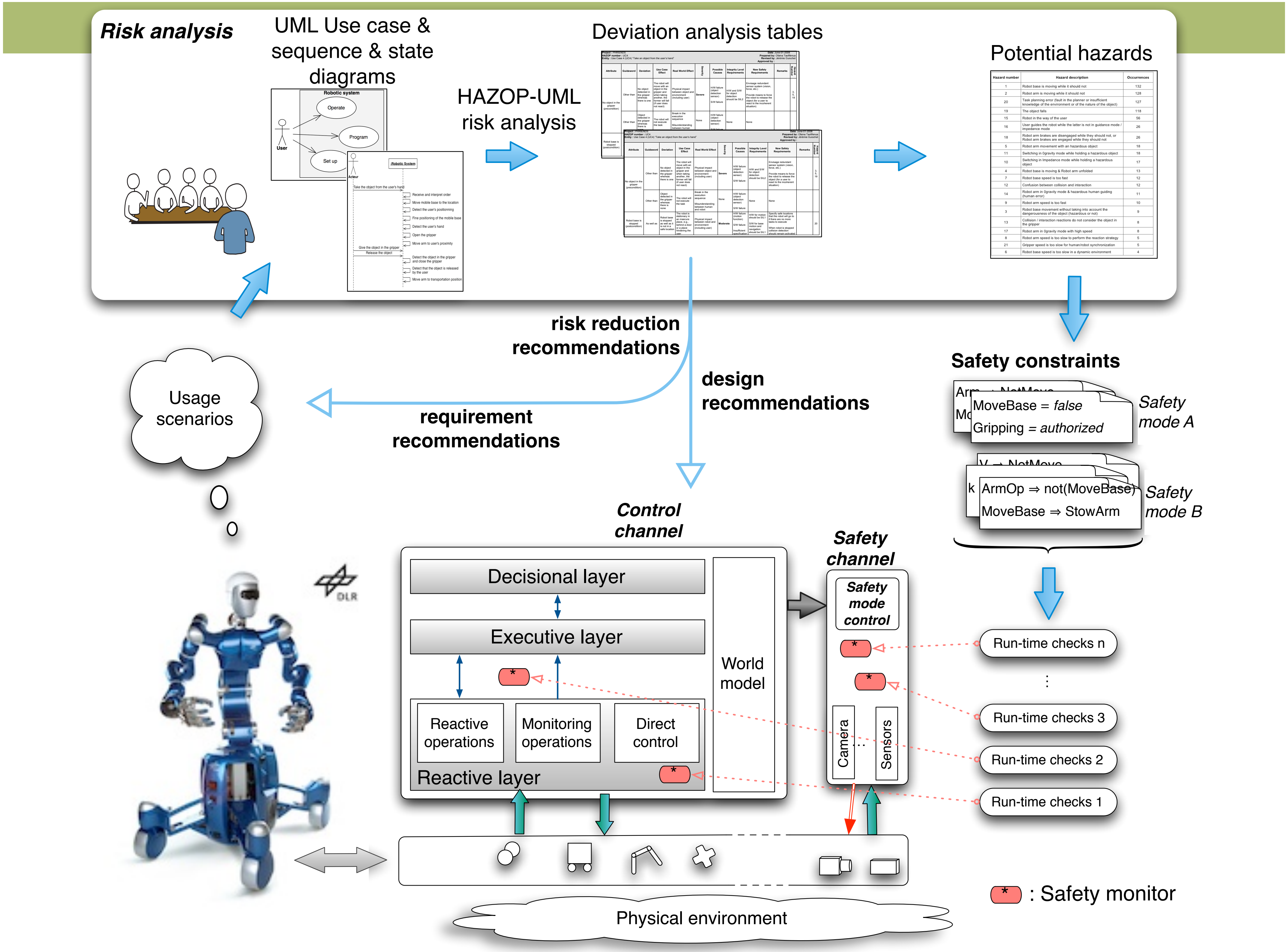
Date: June-01-2008
Prepared by: Ofaina Taofifenua
Revised by: Jérémie Guiochet
Approved by:

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

Results for Model Based Risk Analysis

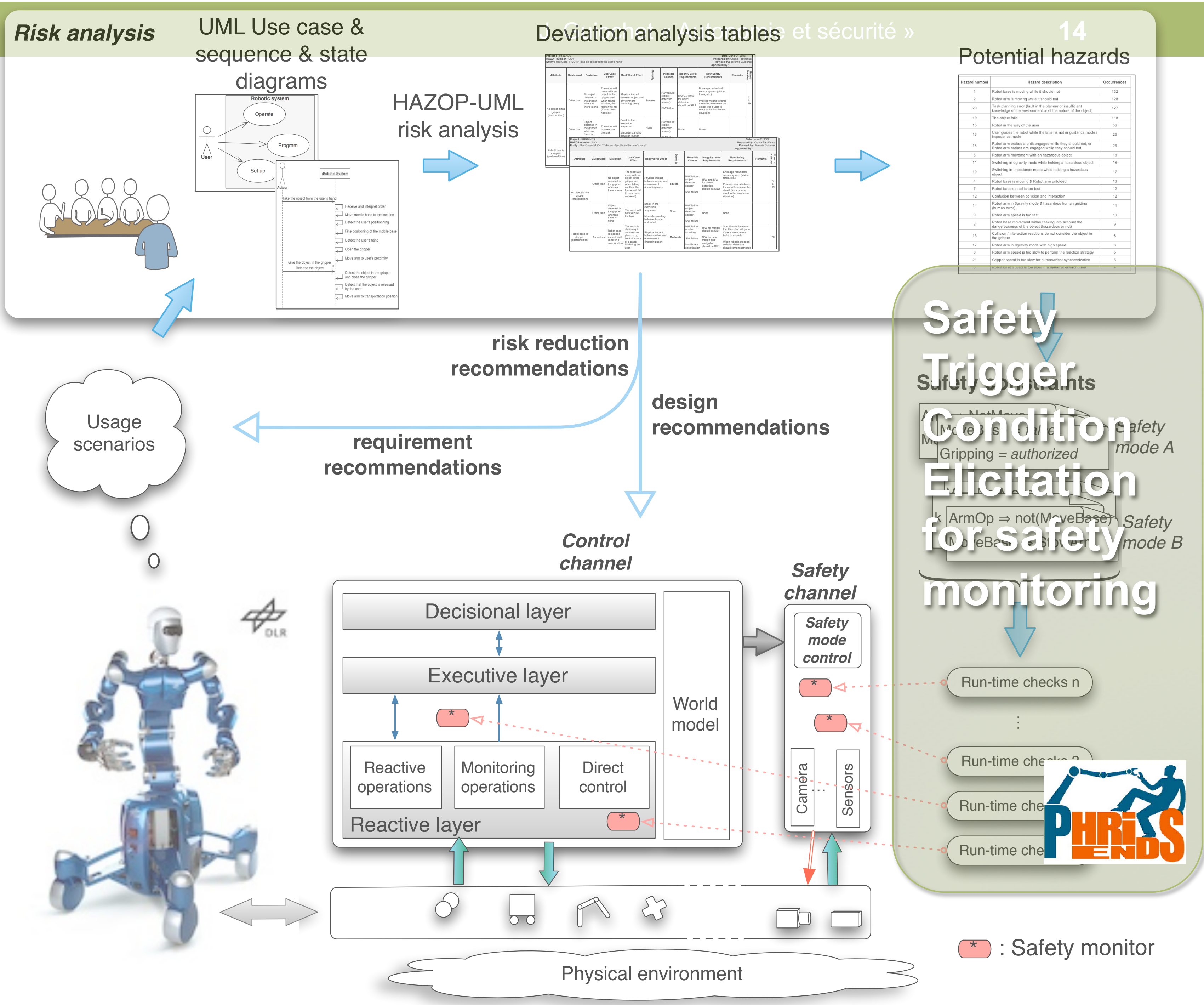
- Applied to
  - an assistive robot for strolling with autonomous navigation (ANR-MIRAS)
  - a co-worker, able to fetch, pick, carry, and give tools (FP7-PHRIENDS)
- Systematic approach, mainly based on scenario description
  - ☺ do not depend on architecture & technologies, focus on interactions
  - ☺ limit combinatory explosion
  - ☺ manage a part of uncertainties
  - ☹ do not include environment adverse situations
  - ☹ strongly based on level of expertise of the safety expert
  - ☹ qualitative and not formal

Provides a list of potential hazards



Risk analysis
UML Use case & sequence & state diagrams
Robotic system
Operate
Program
Set up
User
HAZOP-UML risk analysis
Deviation analysis tables
Potential hazards
risk reduction recommendations
requirement recommendations
design recommendations
Usage scenarios
Safety constraints
MoveBase = false
Gripping = authorized
Safety mode A
ArmOp ⇒ not(MoveBase)
MoveBase ⇒ StowArm
Safety mode B
Control channel
Decisional layer
Executive layer
World model
Reactive operations
Monitoring operations
Direct control
Reactive layer
Safety channel
Safety mode control
Camera
Sensors
Run-time checks n
Run-time checks 3
Run-time checks 2
Run-time checks 1
DLR
Physical environment
: Safety monitor

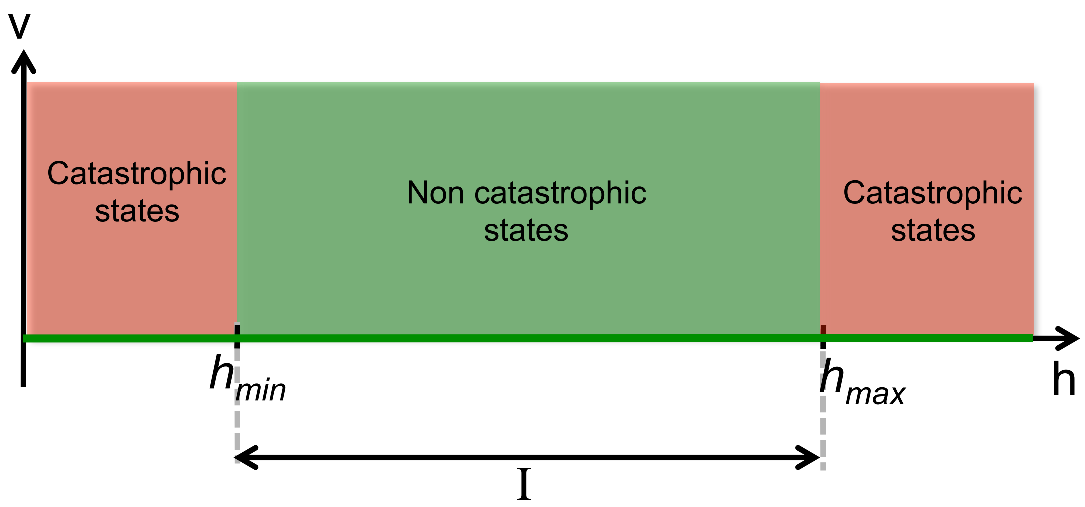
Safety Trigger Condition Elicitation for safety monitoring



Toy example

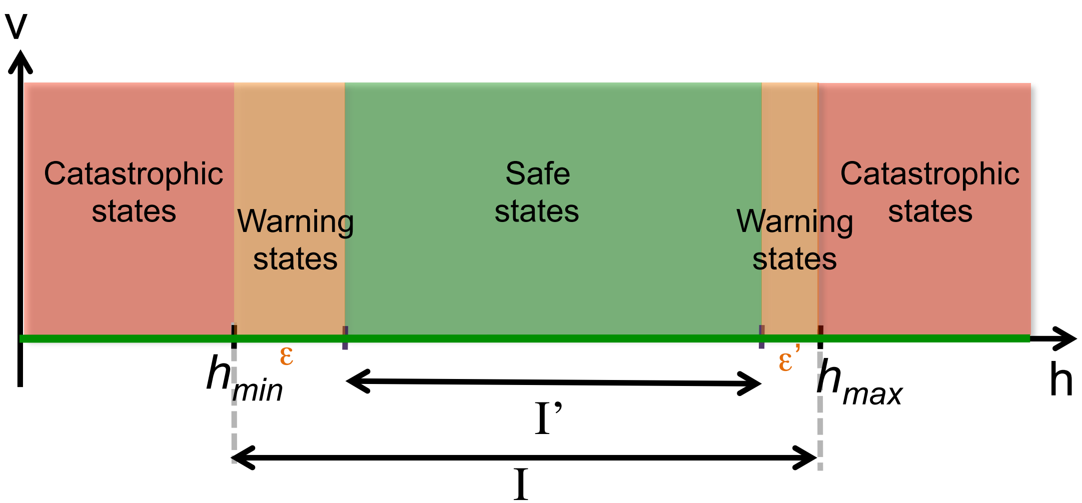
- Hazardous situation : “ The handles are at a bad height during strolling” $(v>0) \wedge (h \notin I)$
- Safety condition can be formally defined by :

$$(v=0) \vee (h \in I)$$



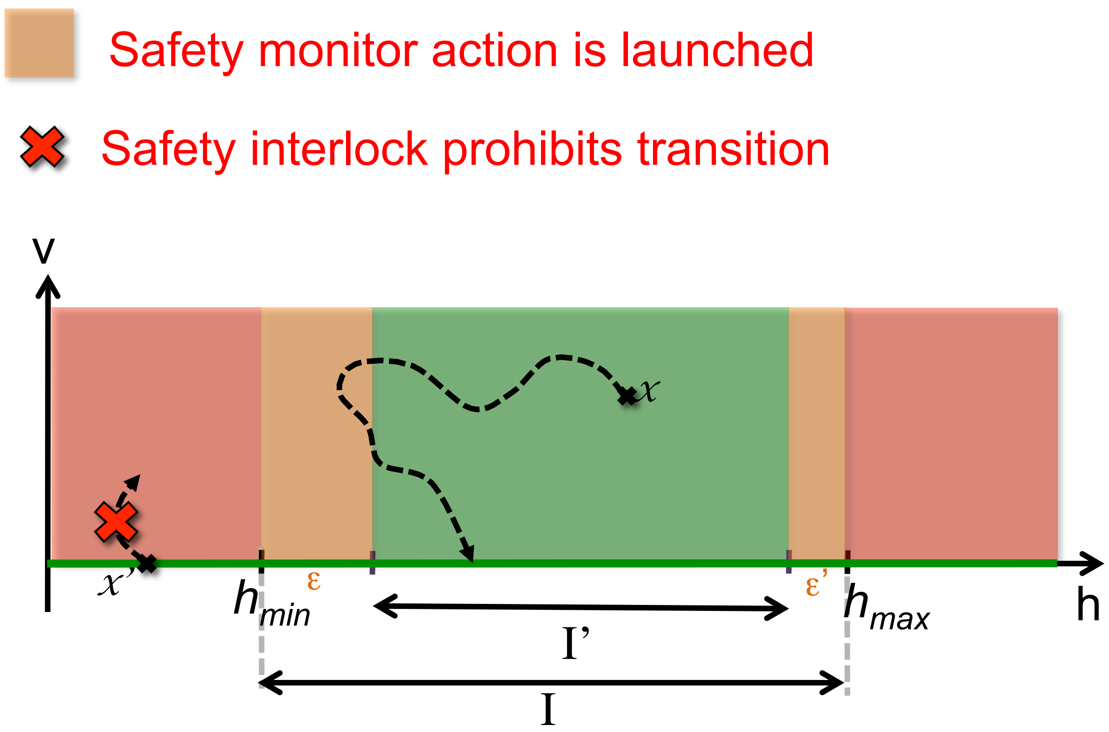
Toy example (2)

- Warning states identification



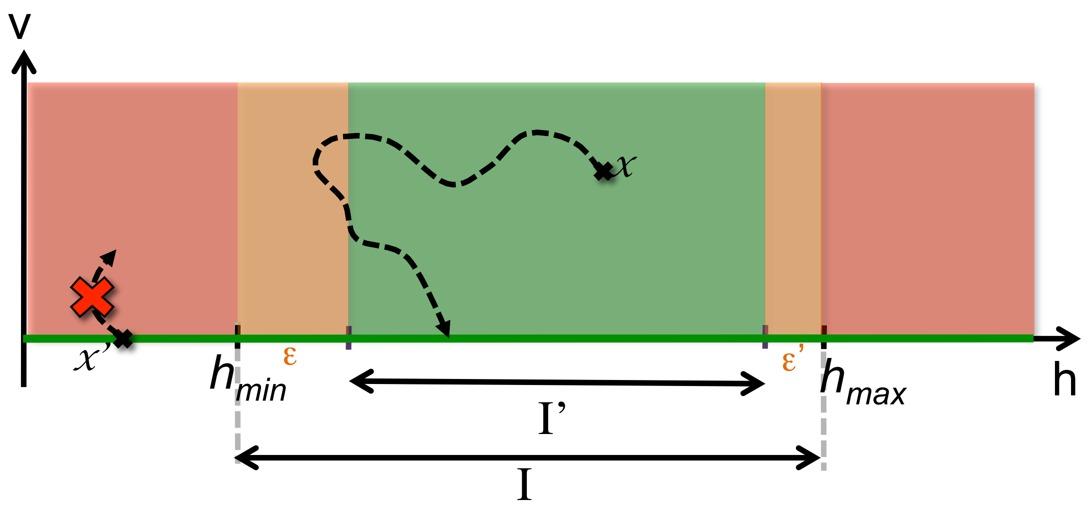
Toy example (3)

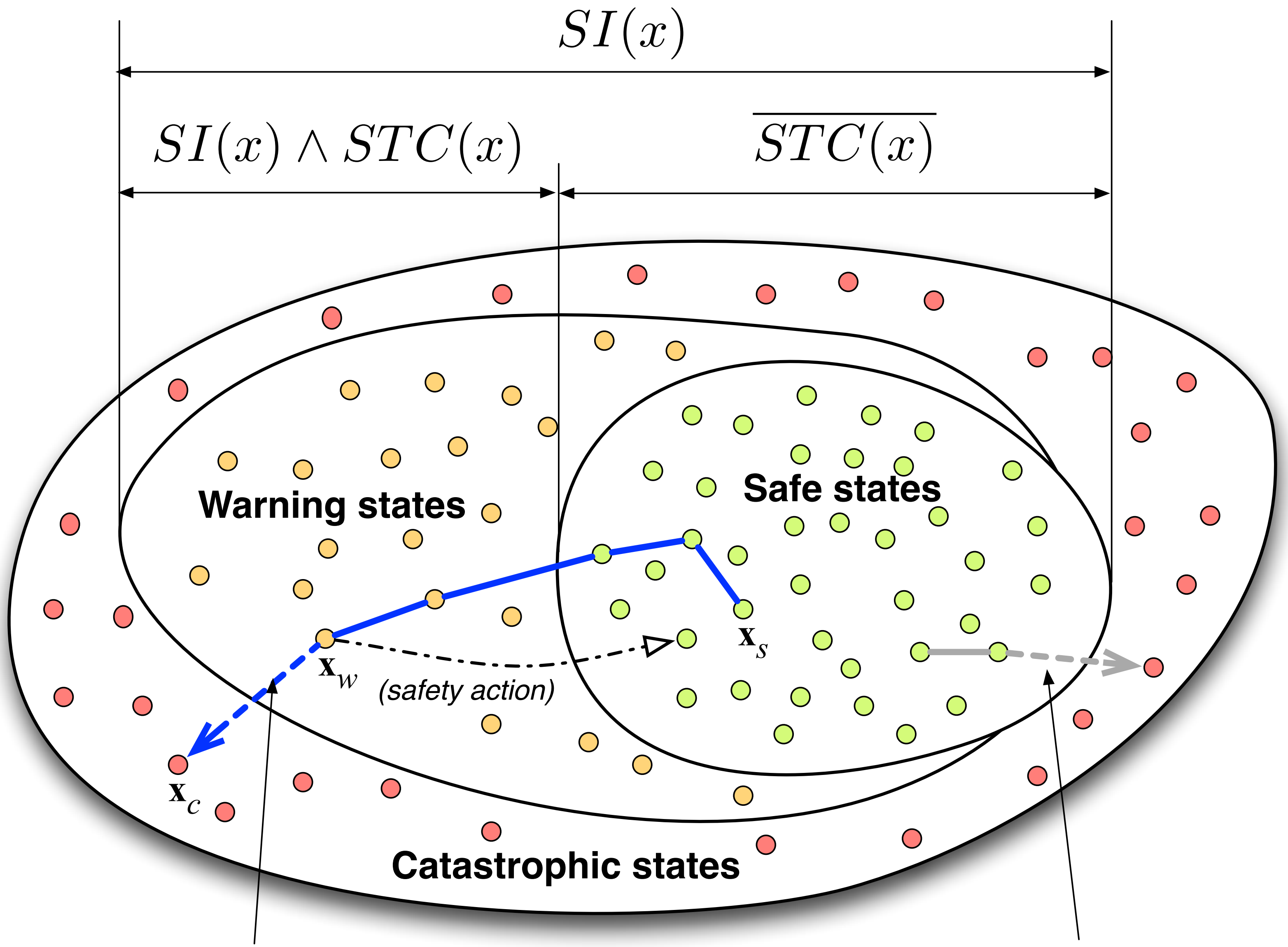
- Safety monitor and interlocks



Toy example (4)

- Safety invariant (SI) and safety trigger condition (STC)
- $SI(x)=((v=0) \vee (h \in I))$
- $STC(x)=((v>0) \wedge (h \in I \setminus I'))$





Path to be aborted by safety action triggered by safety monitor

Path to be removed by a safety interlock

Definitions (if needed)

- **Safety condition** : sufficient condition to avoid a hazardous situation.
- **Safety invariant (SI)** : *necessary* safety condition, i.e., the violation of a safety invariant is intolerable in that it implies immediate harm and violation of a high-level safety requirement.
- **Safety action :** activity carried out explicitly to bring the system to a safe state.
- **Safety trigger condition (STC):** condition that, when asserted, triggers a safety action.
- **Safety margin** : “distance” between a safety trigger condition and the negation of a safety invariant.

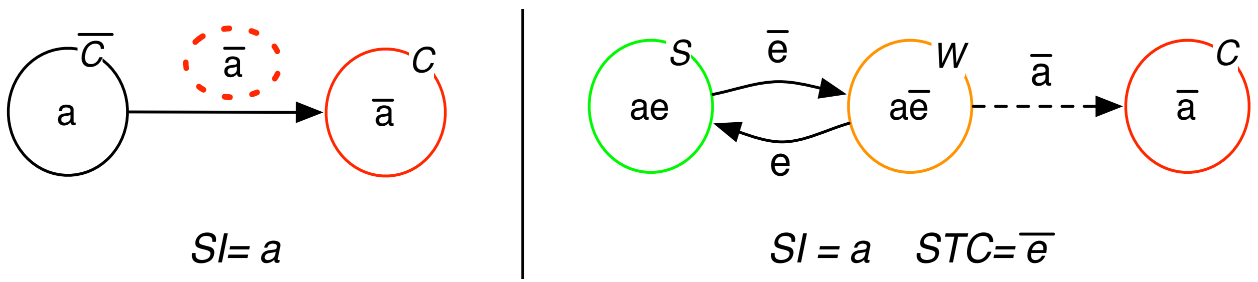
Overview of the process

1. extract sufficient ***safety conditions*** *from* HAZOP/UML risk analysis.
2. for each *safety condition*, define, if possible, a ***safety margin*** on each safety-relevant variable, and thereby, the set of warning states. If a safety margin can not be defined for a particular variable, the safety condition must be enforced by some other mechanism (e.g., a physical interlock).
3. if safety margins and safety actions have been defined, we verify the **consistency of safety action**s that can be carried out simultaneously.

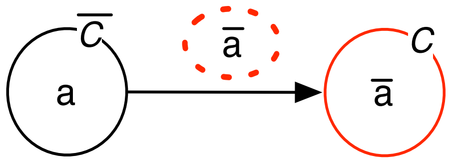
Safety margin elicitation

- Hypothesis
  - Each safety invariant is expressed as a disjonction *of atoms :*

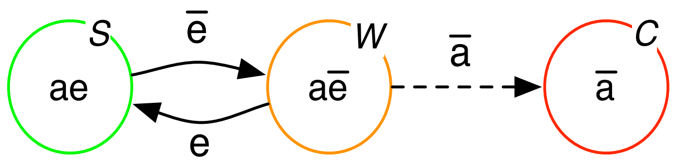
  $SI = a \vee b \vee c \ldots$ *(or* $SI=a$*),* where atoms are propositional variables
  - Atoms are independants (i.e. there is no function between safety relevant variables of two atoms of one SI)
- Margin calculation is done introducing a variable e, that produces a partition of the non catastrophic region
  - Mathematical proof for margin existance and calculation (e->a)



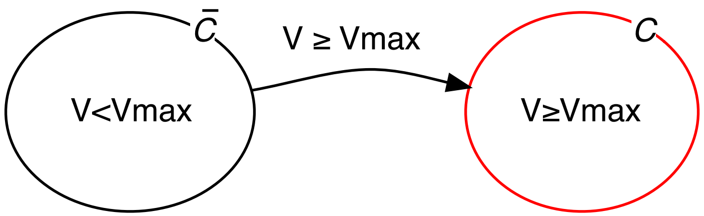
Application – Robot Speed



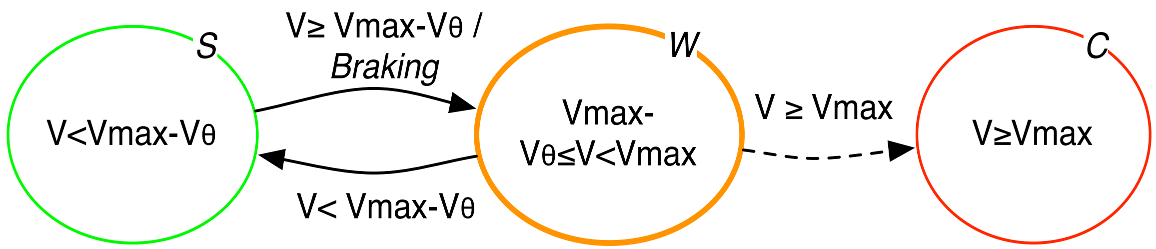
SI= a



SI = a *STC=$\bar{e}$*



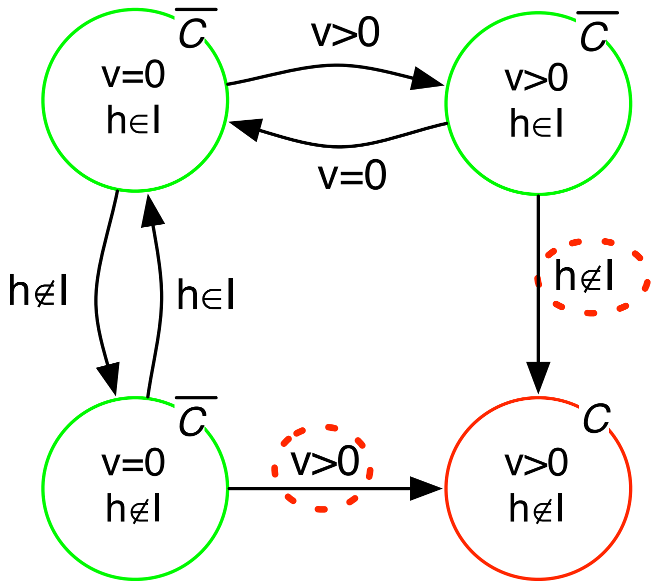
SI = (V<Vmax)

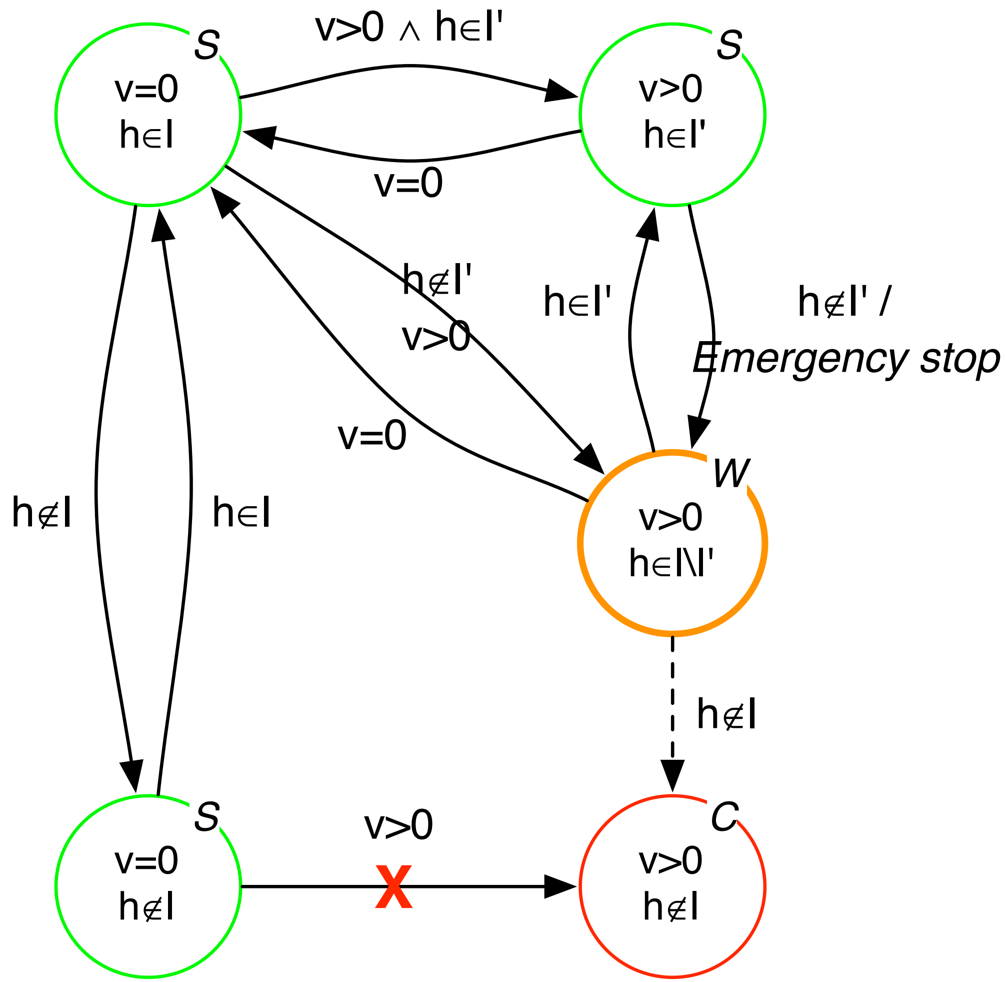


STC = V ≥ Vmax-Vθ

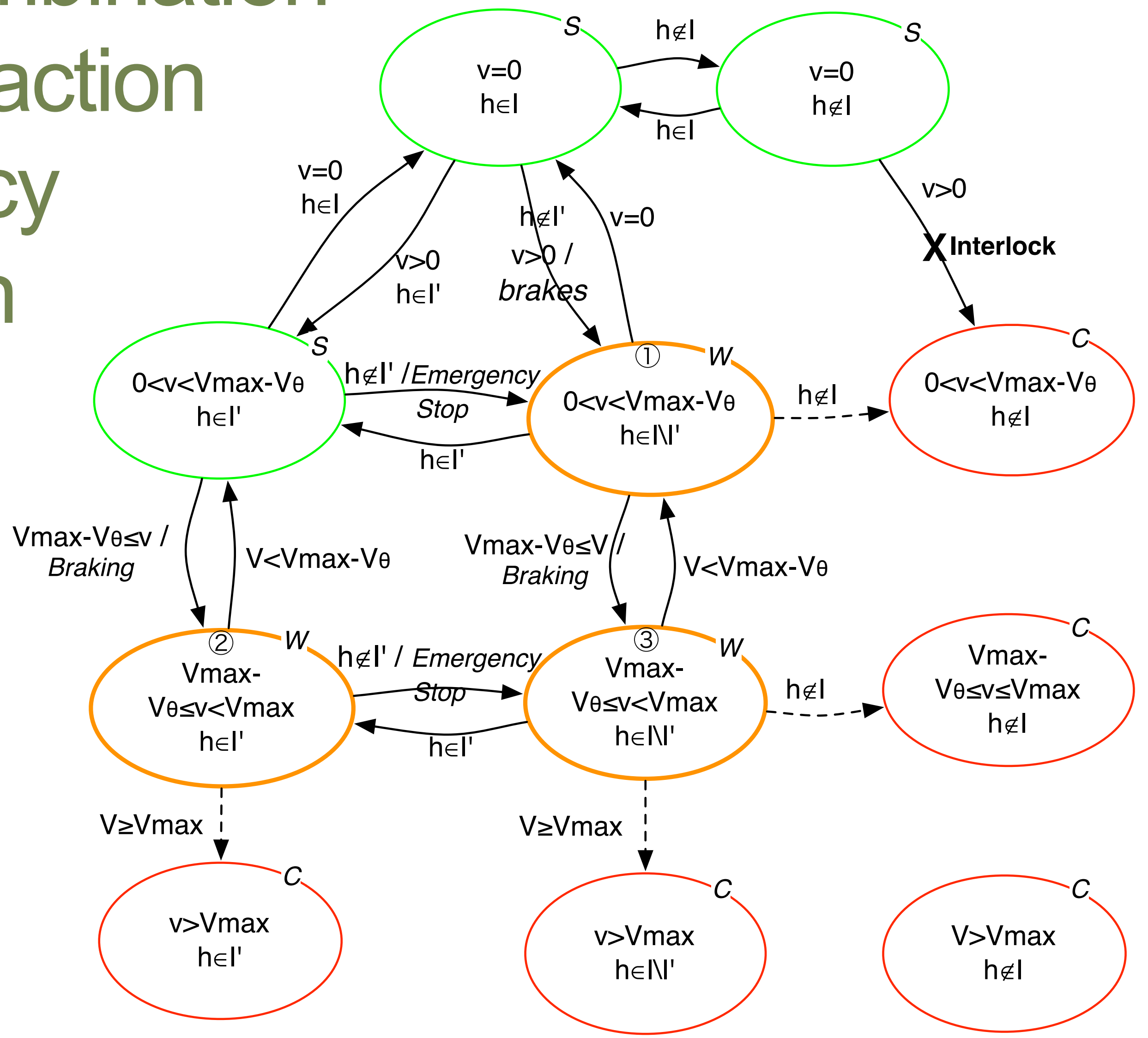
Toy example

- $SI(x) = (a \vee b) = ((v=0) \vee (h \in I))$





Graph combination for safety action consistency verification



Results for online safety monitoring

- A collaborative method for safety trigger condition and interlock elicitation
  - Collaborative : between safety analysts and domain experts
  - Consistency between STC and interlocks (often not checked)
  - Manage complexity (divide to reign), ready for application with many and complex safety invariants (for complex tasks in non structured environment)

- Next steps
  - Some mathematical proves TBD
  - Consistency of safety actions
  - Tool for calculating margins and interlocks
  - Safety monitor prototype
  - Part of multi-level safety monitoring