Architectural patterns for functional safety

Jari Rauhamäki[1], Timo Vepsäläinen[1], Seppo Kuikka[1]

[1]Tampere University of Technology, Department of Automation Science and Engineering
P.O. Box 692, FI-33101 Tampere, Finland, {jari.rauhamaki@tut.fi, timo.vepsalainen,
seppo.kuikka}@tut.fi

1 Introduction

Safety is an emerging issue that is constantly gaining more importance in many sectors including industrial control and machinery applications. Safety is required by laws and regulations and demanded by customers. Consequently, vendors are required to offer safety certified products. As safety becomes more mainstream, a cost-effective safety system design process gives an edge to a vendor. Design pattern approach can help to simplify the design process and provide an easy to understand view to safety-related systems.

Design patterns are popular in the field of software engineering and there are plenty of patterns. Contradictorily, in the field of control and safety engineering patterns have not been studied and published in such volumes. We have now answered the call. During our studies on safety-related software applications in a machinery domain, we have been able to identify some design patterns related to safety control applications.

This paper presents four patterns related to the development of machine and industrial process control applications. The patterns have not been, exactly, mined from industrial applications developed by companies because the access to documents is restricted. However, according to studies and interviews with professionals in the industrial control domain, the solutions that the patterns describe are known in the industry and utilized in the industrial control domain in both machinery and industrial process control.

1.1 Safety-related system

The patterns in this paper focus on safety-related systems. Before we can define a safety-related system, we need to define the term safety in general. The IEC 61508 [1] (in the part 4) states that safety is “freedom from unacceptable risk”. This is a generic definition covering all kinds of specific definitions related to e.g. physical, financial or social damage or hazards [2].

A safety-related system can now be defined as a system that “implements the required safety functions necessary to achieve or maintain a safe state for the EUC

(Equipment Under Control)" and "is intended to achieve, on its own or with other safety-related systems, the necessary level of safety integrity for the implementation of the required safety functions." [1]. Thus, safety-related system reduces the risk of an undesired event on acceptable risk level by affecting the operation of a system. A safety-related system may reduce the risk of a hazard by reducing either the probability or the consequences of the hazard, which are the factors of the risk.

1.2 Safety-related system development

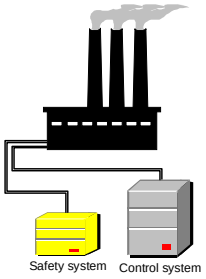
In principle, a safety-related system can be considered just another system though it has a specific task in the system. Thus, any system that is indented to maintain the safety of the system and reduce risk to a tolerable level is a safety system. Unfortunately, the situation is not as simple in the real world. As the safety-related system is the final frontier to prevent the realisation of a risk, the safety-related system development is heavily regulated. Regulations, laws and standards present requirements for the safety-related system development.

Typically safety-related systems are developed to be compliant with standards regulating the devices and machines of the considered domain. The safety-related system must take into account the requirements proposed by the standard. For instance, the IEC 61508 is a generic standard for the development of safety-related systems whereas the EN ISO 13849-1 [3] is focused in safety of machinery applications. The standards define a set of methods and techniques to be used in the development process. In addition, the standards propose requirements on the structure and the operation of the system. The IEC 61508, for instance, defines how a safety-related system must be developed. In contrast, domain specific standards are typically more concerned with the safety-functionality of the system. That is, what kind of safety functions the system must implement (emergency power off, a shutdown if people enter working area, etc.).

2 Separated safety

Context

A control system for a work machine or an industrial process needs to be designed and developed. According to performed hazard and risk analyses, the system to be controlled is capable of causing physical or economic harm to the environment or people working in its surroundings. Because of the possible risks, the functional safety of the system must be ensured with a safety system that must be developed according to appropriate standards and possibly certified by authorities.



Problem

How to avoid the situation in which the whole control system would need to be developed according to the IEC 61508 or other safety standard?

Forces

- Safety: the functional safety of the system must be assured with an appropriate solution.
- Standards: Safety-related standards such as IEC 61508 [1] require independence between safety-related and non-safety-related systems
- Cost-efficiency: Development of the whole control system according to safety standards would be difficult – if not impossible – and increase the development costs substantially.
- Cost-inefficiency: Use of certified components in the whole control system would increase the hardware costs substantially.
- Suitability: Certified components and processing units with limited instruction sets may not enable development of all required control functionalities. Limitations apply to, for example, floating point arithmetics.

Solution

The problem is solved by dividing the control functionality into two separated systems: basic control system and safety system. Requirements for the whole control system are first divided into safety-critical (SCR) and non-safety-critical requirements. Typically, the safety-critical requirements are related to deviation and possibly hazardous situations whereas the non-safety-critical requirements are related to normal operational conditions and the intended use of the system. Safety-critical functionality is then designed and implemented into a safety system according to safety standards. Non-safety-critical functionality is designed and implemented into a basic control system. **Fig. 1** illustrates the process of separation of the systems.

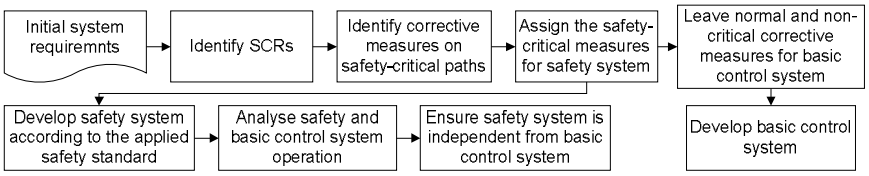


Fig. 1. Simplified design flow for separation of safety system from basic control system

The safety system and the basic control system are separated from each other so that the correct functioning of the safety system is not dependent of the correct functioning of the basic control system. If necessary, the safety system may utilize certified hardware such as sensors, actuators, buses and safety PLCs. The basic control system may utilize the same components provided that it is not capable of disturbing the correct functioning of the safety system; otherwise, it must use different components. Because the basic control system is separated from the safety system, the requirements of safety standards do not apply to the development of it.

Consequences

+ Safety of the system can be achieved with an appropriate safety system.
+ Basic control system development may utilize the development process, tools and techniques preferred by the company – not the ones required by safety standards.
+ Full instruction set tools, computing units and components can be used with the basic control system.
+ As the safety is ensured with a separated system, the basic control system does not need a certification.
+ The development costs of the basic control system can be reduced.
+ Because the safety and basic control system are separated, the development of them can be outsourced separately or they can be developed independently from each other by different development teams. This can also affect positively to the schedule of the whole project.
- Two separated applications must be developed and they may require different instrumentation.
- Increased cost due to development, instrumentation and maintenance of an additional system.

Resulting context

The resulting solution consists of two separated systems which can be developed separately so that the independency of the safety system from the basic control system can be proved to the authorities.

Related patterns

Productive safety describes how to divide the responsibilities so that the economic consequences of the activations of the safety systems can be reduced to only necessary situations.

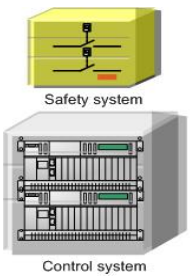
Known usage

During various research projects, the researches of the Automation software engineering research group (AOT, department of Automation science and engineering) have participated in the interviews of the professionals of industrial control and mobile working machine sectors. According to the interviews, this solution is widely used in Finnish process industry and also known in the domain of mobile work machines.

3 Productive safety

Context

A control system for a work machine or an industrial process is being developed and the *Separated safety* pattern has been utilized so that the control system functionality is divided into two separate systems: basic control system and safety system. However, some user requirements are safety-related although not



safety-critical. Quite often, these kinds of requirements can be due to customers wanting to avoid the economic consequences of the activations of the safety system, such as shutting down the machine or plant. Or, the system should be operable near the safety limits so that the productivity of the system can be increased.

Problem

How to divide the responsibilities between the safety system and the basic control system so that the system would remain operational as long as possible and as near the safety limits as possible?

Forces

- Safety: the safety of the system must be ensured in every foreseeable situation.
- Productivity: operating near safety limits often increases the productivity of a process.
- Economy: economical impacts of, for example, running down a paper machine or power plant are dramatic and not desired unless it is not absolutely necessary to achieve safety.
- Recovery from deviations: the customers and users of the system want the system to try recovering from and correction of disturbances. The recovery algorithms may require complex and advanced functionality and/or logic.

Solution

The corrective functions for disturbances, that are necessary for fulfilling the requirements of clients, are implemented in the basic control system. In this way, the scope of the basic control system is widened to include functions the purpose of which is to keep the system in its operation region in which safety system never activates. However, these kinds of requirements are easier to implement in the basic control system that does not need to be designed and implemented according to safety standards. In the basic control system, the corrective actions (interlockings) can be as complex as required to achieve the goal (**Fig. 2** a).

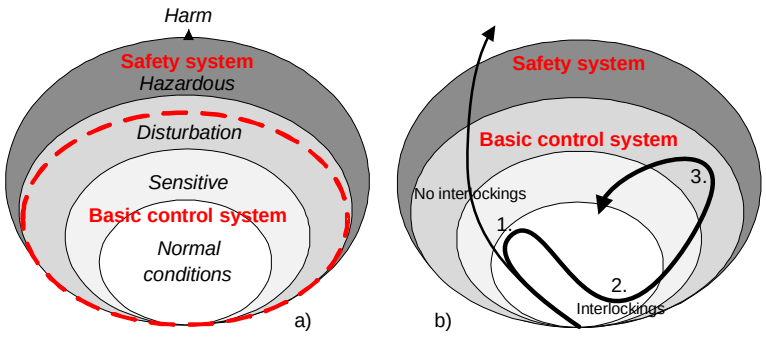


Fig. 2. Layers of operation from control system point of view.

The safety system, on the other hand, can be designed to be as simple as possible. In many cases, the system is in the safe state when the system is not powered. So, without trying to recover from disturbances, the safety system can often be designed to shut down the whole system when critical safety limits are violated.

In **Fig. 2** b) a possible operation path of a process is illustrated. The process is first in normal operation condition. Soon the state starts to shift towards sensitive area. In point 1, the interlockings prevent the process from entering to the sensitive state. If there were no interlocking the state would have changed to hazardous where the safety system had taken control. After point 2 the process state leaves normal conditions and enters the disturbed state. The interlocking didn't manage to return the state to normal in the sensitive state, but the state begins to move towards normal from disturbed conditions in point 3 due to successful interlocking operations. The main idea of the interlockings is to prevent the system entering the hazardous state that would cause the safety system intervention and e.g. complete shutdown of the system.

Consequences

+ The safety of the system can be achieved with a safety system that is designed to be as simple as possible which makes it easier to develop and certify.
+ The corrective actions that are required for operating the system near safety limits can be implemented without the need to follow safety standards. Operating the system near safety limits often increases the productivity of the system.
+ The system is not shut down by the safety system unless absolutely necessary.
- The complexity of the basic control system is increased
- Strict appliance of simplicity in safety system rules out advanced safety functions

Resulting context

The safety of the system is still ensured with the use of the safety system that can be developed to be as simple as possible. The scope of the basic control system, on the other hand, is widened to include functions the purpose of which is to keep the system in a safe region so that the safety system gets never activated.

Related patterns

The Separated safety pattern describes how to divide the system into safety-critical and non-safety-critical parts. This pattern describes how to set the responsibilities of the parts so that the system can be operated near safety limits and in deviation situations without compromising the safety.

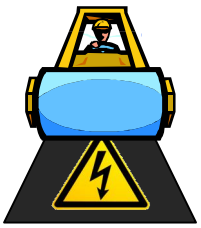
Known usage

Similarly to *Separated safety* pattern, the solution is well-known in the domain of industrial control. In a chemical process, an example of a corrective action of a basic control system could be relieving pressure with a relief valve before actual safety limits whereas a safety system could be designed to stop all heaters and pumps related to the process.

4 Safety overrides

Context

A control system for a work machine or for an industrial process is being developed and the *Separated Safety* pattern has been utilized so that the control system functionality is divided into two separate systems: basic control system and safety system. The separated systems may in some places control the same functionalities or process variables.



Problem

How to ensure that the safety system can always override the basic control system?

Forces

- Safety: Safety control system must always be able to drive the system into safe state (i.e. state in which system minimizes the risk of damaging itself or people around it) regardless of the state of the basic control system
- Reliability: Redundant safety functions increase reliability
- Regulations: Safety standards require safety functions to be prioritized over normal functions
- Cost-efficiency: Additional hardware increases costs

Solution

Provide the safety control system with the ability to override the control system's operations. Ensure that the safety function cannot be circumvented or bypassed in any way by the control system. To implement the override capability there are three main approaches. In each approach, the safety system has ability to drive the state of the controlled quantity to the safe state regardless of the control system state. Notice, that safety function typically wants to either fully enable or disable controlled variable. The safety actuator is then placed in parallel or in series in terms of the control actuator respectively. The three approaches are described in the following list.

1. The first approach is to provide the safety system with a separate actuator that is able to drive the system into the safe state. The safety actuator must be placed so that the normal control system cannot bypass it.
2. The second approach is to use a shared actuator between safety and control functions. In this approach additional logic is required (either separated device or build in the shared actuator). The logic takes inputs from the safety and control systems. The logic operates in such way that it prioritizes the output of the safety system over the control system. One actuator is spared, but additional logic (or typically more expensive actuator hardware) is required.
3. The third approach is a combination of the first two approaches. The safety system controls a dedicated actuator and is able to override the control of the control actuator. The approach employs redundancy, which promotes reliability (of the

safety function). If one of the actuators fails, the other one is still capable of applying the safe state.

Example

Typical application of the safety overrides principle can be found for example in processes in which the flow of steam in a pipe is controlled (see **Fig 3**). The control system is responsible for controlling the flow using a proportional valve. In addition the steam line is equipped with a safety valve controlled by the safety system. Now, regardless of the basic control system the safety control system may halt steam flow in the pipe.

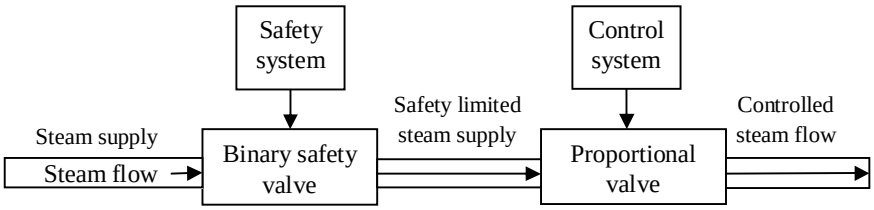


Fig 3. Example of safety overrides in steam flow control

Consequences

+ Safety is retained by safety system if control system fails
– Additional hardware/logic adds the total cost of the system
– Additional hardware/logic adds the complexity of the system

Resulting context

The safety control system may override the control system in all situations and thus safety is not dependent on the control system.

Related patterns

Hardwired Safety pattern describes how a safety system can be implemented without software.

5 Hardwired Safety

Context

There is a system that is controlled with separated safety and normal control systems, i.e. *Separated Safety* pattern is applied. The system resembles more a unique project than a mass product and safety functions are considerably simple.

Problem

How to implement safety functions without safety-critical application software?

Forces

- Cost-efficiency: Development of safety-critical software is costly and time consuming and unnecessary safety controller are expensive
- Complexity: Safety systems should be simple and understandable
- Maintainability: Safety systems should be easy to maintain

Solution

Instead of software-based solution, use a hardware based safety system. Hardwired safety systems can be used to implement simple and generic safety functions such as over and under temperature, pressure and speed related to a process variable. Certified COTS hardware for such generic functions is available. Advanced and custom safety functions are, however, easier to implement with custom software based safety applications.

Remove the need for safety-controller and safety-critical application software[1] by establishing direct link between sensor and actuator hardware. The safety system consists of a data source (sensor) and an actuator. The sensor measures system state and the trigs the actuator to apply safe state when defined conditions apply. The devices need to compatible in terms of communication. That is the sensor must provide suitable output signal and the actuator needs to be able to use the signal generated by the sensor.

The following guidelines can be used to identify safety functions, which could be implemented with hardwired solutions.

- Firstly, the safety function requires no complex logic or calculations. In principle, any logic can be implemented with simple hardware devices (such as logic gates). However, in practice merely simple and/or functions are sensible as hardware implementations in small amounts.
- Secondly, there should be a well-defined trigger for the safety function. In this context a trigger means an event in the system that trigs the safety function active (e.g. liquid level rises above a maximum value). Simple logics can be used to connect several trigger conditions (e.g. liquid level high and exhaust valve closed). However, advanced conditions are problematic, e.g. mean value.
- Thirdly, the safety function should be able to actuate the safety function in relatively simple manners. That is, positioning multiple outputs to arbitrary states is problematic whereas controlling single binary output is considerably easier.

Example

A possible utilization target for hardwired safety function would be, for example, the over temperature protection function of a heating container. In the requirements for the safety of a plant, a requirement for an over temperature limiting of the main heating container of the plant is given. As the plant development is (at least nearly) a

[1] In this context safety related custom application software refers to software that is developed for the system under design. Embedded software (e.g. firmware) in safety certified COTS devices is not counted as (custom safety related) application software.

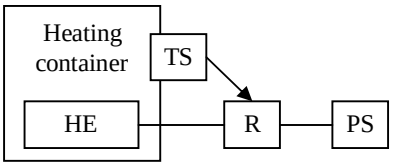


Fig. 4. Safety function with hardware implementation

unique project, hardwired safety systems are used to minimize safety related application software. The solution is depicted in **Fig. 4**.

A temperature sensor (TS) is applied to the container being monitored for over temperature. A relay (R) is applied between power source (PS) and heater element (HE) of the container. The sensor output is connected to relay input and when the temperature reaches a predefined value, the output of the sensor goes off and relay opens. The power source is detached from the heating element and temperature of the container no more rises. A safe state is obtained.

Consequences

+ Safety related custom application software is reduced
+ No need for dedicated safety-related controller
+ Time and money is saved during the design process
+ Easy to understand safety implementation is achieved
– Advanced features of the safety system are hard to implement with pure hardware solutions
– Expansion and further development of hardwired solutions are harder than software based solutions

Resulting context

Safety function is implemented with minimum amount application software. Safety functions are implemented by a hardwired safety system.

Related patterns

The operability of the system can be improved utilizing *Productive safety* pattern. Advanced safety functionalities can be implemented using software.

6 References

[1] IEC 61508. Functional safety of electrical/electronic/programmable electronic safety-related systems. *International Electrotechnical Commission*. 2010.

[2] Wikipedia. Safety. http://en.wikipedia.org/wiki/Safety [last modified on 20 November 2011]. Accessed 2 December 2011.

[3] EN ISO 13849-1. Safety of machinery, Safety-related parts of control systems, Part 1: General principles for design. International Organization for Standardization. 2006.