COMPUTER-AIDED SAFETY ANALYSIS OF COMPUTER-CONTROLLED SYSTEMS: A CASE EXAMPLE

UWE BIEGERT

University of Stuttgart, Institute of Industrial Automation and Software Engineering,
biegert@ias.uni-stuttgart.de, http:\\www.ias.uni-stuttgart.de

Abstract. Computer controlled systems consist of a complex interaction between technical process, human task and software. For the development of safety critical systems new method are required, which not only consider one of these parts of a computer-controlled system. In this paper a qualitative modeling method is presented. The method is called SQMA, Situationbased Qualitative Modeling and Analysis and it origin goes back to Qualitative Reasoning. First, all parts of a system are modeled separated and then combined to a unique model of a computer-controlled system. With this qualitative model a computer supported hazard analysis can be realised.

Key Words: safety analysis, qualitative modelling, computer controlled systems, ROOM

1 INTRODUCTION

In recent time, it is recognized that, especially for the safety analysis of computer-controlled systems, new common methods are required. Nowadays software makes it possible to control more complex processes, but at the same time it is responsible for the welfare of humans and environment. Failures in a software program can influence the technical process with unforeseeable effects. On the other hand, a defect in technical components may influence the regular behavior of a control program. Furthermore human tasks effect the normal flow of the process either in a direct way or in an indirect way by the control program.

Many catastrophes show clearly that accidents do not only rely on an single failure but also on a combination of failures (multiple failures). Evaluating the consequences of multiple failures conventionally is hard or almost impossible. Many exiting methods for safety analysis are only a manual for a systematic proceeding. The analysis is performed in an expert's mind (brainstorming) and solely relies on expert judgement, experience and knowledge. Frequently the complex interaction between the system components overextend an expert to judge whether a system is safe or not. Not rarely complex interrelations between system components lead the experts to the limits of their abilities.

Within new approaches a computer-aided safety analysis can be realized. The combinatorial thinking is more convenient and is done faster by a computer program, similarly to a chess program. An obligatory prerequisite for the implementation of a computer-aided safety analysis is a description of the controlled system, which is interpretable by the computer. But even this prerequisite represents the heaviest burden. Nowadays there is still no suitable modeling method, which is able to describe the behavior of the technical process, the automation software and all the possibilities of the human task simultaneously [1,2]. One of the big problems that exists is the complexity of such systems. In general it is almost impossible to describe the whole system and the relations of their components with formulas. Furthermore and especially for safety analysis, it is not important to model each detail of the system.

To build such models using qualitative modeling is a new approach. But what is qualitative modeling?

To demonstrate the idea of qualitative modeling we consider a little example (Fig. 1). A human understands the function of the represented system in qualitative way. We know the substantial functions of the components and therefore we can suggest on the possible behavior of the total system. Some events of the system are not important for the behavior. The bullet (E) may hit the balloon or not, the speed and size of the bullet is not necessary for this problem. Some events of the system are not important for the behavior. The bullet (E) may hit the balloon or not, the speed and size of the bullet is not necessary for this problem. We can reduce the interrelation between bullet and balloon to a simple statement: if the bullet hits the balloon, then the balloon explodes (and the brick falls down). If the bullet misses the balloon, nothing will happen. Exactly this abstracted knowledge about the behavior of the system is transmitted to the computer by a qualitative model.

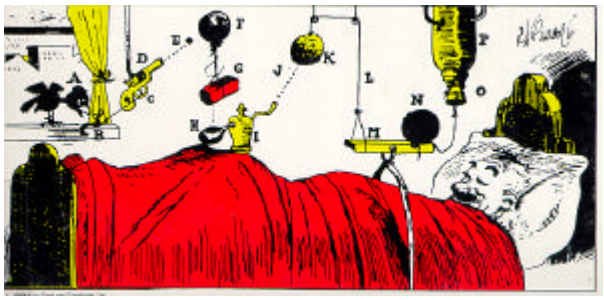

Fig. 1. Example of a complex system[1]

In the scope of this paper the method of SQMA (Situationbased Qualitative Modeling and Analysis) is presented. In the first part the principles of SQMA are demonstrated and in the second part an example serves to show its typical application.

2 SQMA

SQMA is a component-oriented modeling method. First, all components of a system are qualitatively described independent of their function later in the system. The structure of the system represents the rules of the interactions between the components. Within this assumptions a model of the whole computer controlled system can be calculated by the computer.

2.1 Modeling Components

SQMA uses a two step technique to model a component of the system:

1. Black box – modeling
2. White box – modeling

In the first step, all possible terminals of a component and their quantities are described by qualitative interval variables. For example the pistol in Figure 1 has two terminals, one for the trigger and the other for the muzzle. The terminal "trigger" has a quantity, which determines the necessary pressure to trigger a shot (e.g. more than 5mbar). With interval variables we only have to consider some important values for the pistol's variables. The Quantity "trigger" has the ranges [0, 5mbar]; [5mbar, 3 bar); (3 bar, ∞). The fist interval stands for "not enough pressure", the second for "enough pressure"; and the third for "to much pressure". The quantity of the terminal muzzle has the intervals [0,0] which means "no shot" and [1,∞] stands for "shot".

To make the model easier to read we can add comments to certain intervals like we did above. With the specifications of the terminals and the quantities the computer creates a complete situation space of a component. Situations mean a certain combination of the interval of the quantities. Based on the declarations above, for the pistol there are 3 x 2 = 6 situations possible, Table 2 shows a situation table of the component "pistol".

Table 1. A situation table of the model "pistol".

No.	Trigger	Muzzle
1	Not enough pressure	no shot
2	enough pressure	no shot
3	to much pressure	no shot
4	not enough pressure	shot
5	enough pressure	shot
6	to much pressure	shot

But not all of these combinations are realistic for the behavior of a normal pistol. If the trigger is pulled to soft, the pistol won't shoot. Therefore a secondary step of modeling is necessary to describe the real behavior of the component: the white box – modeling.

The relation of input and output quantities is specified by so called situation rules. For the pistol we can formulate the constrain:

if not enough pressure then pistol won't shoot

or with a simple formula:

```
Trigger < 5mbar -> shot = 0
```

All the situations that don't fulfill this rule are deleted from the situation table (e.g. situation No 2 in Table 2). After considering all rules only situations are left which describe the real behavior of a component. Using comment rules heuristic knowledge is merged to the model of the component. Certain groups of situations are indicated regarding their function or they can be classified regarding their importance for safety. Let's consider the pistol again. What happens if the trigger is pulled to hard. Will it break? Will the pistol shoot anyway? Perhaps the pistol will shoot anyway, then the comment rules is:

Trigger > 3bar => Pistol_shoots, Pistol_is_damaged *D*

All situation with the valve of quantity "trigger" is higher than 3 bar will receive the comment "Pistol_shoots, Pistol_is_damaged, and they are classified with the letter "*D*" (like "damaged").

After we consider the static behavior of the component, we are going to consider the dynamic behavior. Possible transition between the situations depend on many criteria. For example physical quantities rely on certain regularities such as steadiness or differential dependencies. Similar to the situation rules, we use transition rules to specify such regularities.

After having modeled each component, we have to describe the connection between them.

[1] From D.S. Weld and J. de Kleer: Readings in Qualitative Reasoning about Physical Systems, 1990

2.2 Modeling the Connection between Components

Now, we have to think about the interrelation between the components. Therefore a block-oriented view of the system is used. A "net list" describes the structure of the system: Fig. 2 show the example of the component pistol and the component balloon. The net list is:

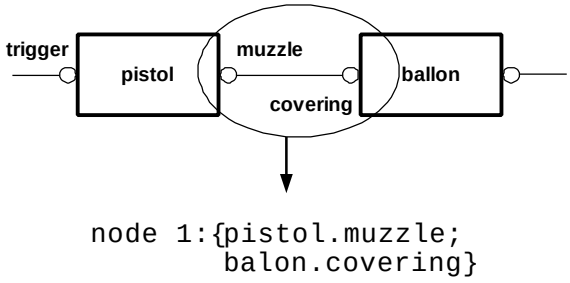

Fig. 2: Building a netlist

Now a program transform the structure of the system into a system of equations interpretable by the computer. The kind of equations depend on the part of the system we are considering. The interrelation of technical components depend on physical regularities, but interrelations between software components depend on special rules of special communication (e.g. bus protocols).

2.3 Analysis of the System

Finally an algorithm combine all situations of the components model. The algorithm considers questions like the following: Is it possible that, component "A" can be in situation "n" and at the same time component "B" is in situation "m". For each combination, the quantities of a component have to be checked with the system equations. If the result of all system equations for a special combination of situations (system situation) is true, then this system situation can happen in reality. Then an even more important question must be answered. Does the system contain dangerous system situations? – When modeling the component we classified different kinds of situations with an attribute. With these attributes a computer can quickly separate the system situations into dangerous, not-intended or regular. Now an analyst doesn't need to think about possible combinations. He only considers the result of the computer analysis. The analyst has to interpret the result himself by considering the likelihood of these system situations and whether prevention is necessary or not.

3 THE EXAMPLE SYSTEM "HOMOGENIZING PLANT"

3.1 The System

Fig. 3 shows a technical system, in which a homogenizing process is to be run. An inlet valve supplies a liquid into a tank with an integrated mixer. The liquid is to be homogenized and to be removed afterwards by the drain valve. The tank has a door, which is to be locked during the process cycle.

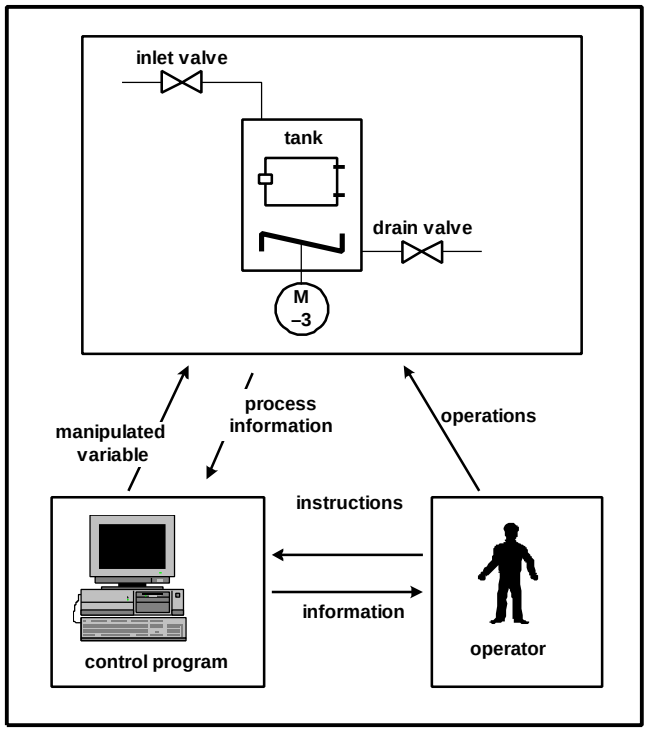

Fig. 3. A computer-controlled system: homogenizing plant

The system was automated with a open loop structure, therefore a operator has control functions like

- starting, terminating and interrupting the process
- modifying the value for the level of the tank (desired value specification)

The function of the controller consists of the following:

- receiving the requests of the operator, checking them and thereupon executing the suitable messages
- The controller must prevent safety critical states of the technical process.

The function of the controller was implemented by a small software program in C++ .

Now, with the help of the technique of SQMA this system is modeled. It will demonstrate how complex the interaction between technical process, automation software and human control interventions turned out even for such a small system. First, all parts of the homogenizing plant are regarded individually and are modeled. Subsequently, the three resulting models are joined to the unique model of the automation system "homogenized plant". With the analysis of the unique model, predicates about the behavior and about the security of the system can be made.

3.2 Qualitative Modeling of the Technical Process

The process is a typical continuous flow process. The technical system consists of two identically constructed electromagnetic valves and a tank with an integrated mixer. The technical system is usually described with PI[2]- and process flow diagrams, which serve as a base for the qualitative modeling of the technical process. In the following, the valve serves as an example for modeling a technical component.

As described in chapter 2.1, the valve is firstly regarded as a black box. The quantities at the terminals are described qualitatively (Figure 4). On the terminals "IN" and "OUT" there appear flow quantities, which consist of the flow strength (I_{IN} , I_{OUT}) and of an appropriate pressure (P_{IN}, P_{OUT}). The terminal C has a manipulated variable "S" (signal), which modifies the status of the valve. These quantities are described with qualitative interval variables. With different intervals of these quantities, we can determine whether the flow is from left to right, vice versa or if there is no flow through the valve.

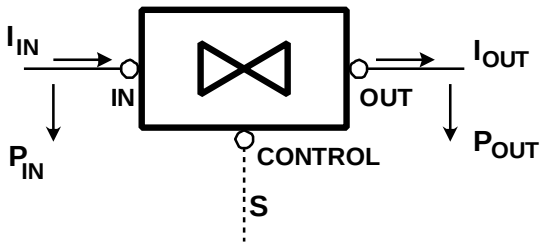

Fig. 4. Black-box view of a valve

In order to model the behavior of the valve, we need to find the appropriate situation rules. The rules for the functioning of a valve are:

- If the manipulated correcting variable "S" is set (S =1, valve opened), then the pressure between the Terminals IN and OUT is equal to zero
- If the correcting variable "S" is not set (S=0, valve closed), then no flow takes place through the valve

Theoretically with respect to the model specification with SQMA, 18 different situations are possible after the black-box step. Only 6 situations fulfill all the situation rules. The remaining situations describe the behavior of the valve. Also, an expert can bring in his knowledge. For example, all situations at which the manipulated variable "S" is set (S =1) and no flow through the valve is possible, receive the comment "defect" and the attribute N (not intended).

The component *mixer tank* is modeled in a similar way. The composition of the three models inlet valve, tank and drain valve take place with the help of a program automatically. Situations of the different models are combined among themselves and are checked for their validity with the system equations, as mentioned in chapter 2. As results we receive a situation table, which contains all possible system situations (altogether 50) of the technical process.

3.3 Qualitative Modeling of the Controller Software

For this system the controller software was developed with the method ROOM. ROOM stands for Real Time Oriented Modeling and supports the development process of software projects [8]. With ROOM functions of the software are divided into components, the so-called actors. These actors can only communicate among themselves by messages on defined channels. Messages have a signal part and optionally a data part. For further information about ROOM, the book "Real-time Object-Oriented Modeling" is recommended [8].

The behavior of the actors is described with so-called ROOM-Charts. They are extended state machines, based on Harel's Statecharts. A transition is triggered by a certain event (signal). Optionally, further conditions for a transition can be specified, e.g. the data part of a messages can be checked for certain values.

Further detail of the design of the controller software is of no importance. To translate a ROOM-design into a qualitative model the signal and data part of a messages must be expressed by using interval variables. The values of the qualitative variables stand for the different contents of a messages. During the black box modeling, the complete situation space is regarded. The situations describe all the possible combinations of the messages from an actor. In contrast to the modeling of the technical process, there is no physical law for the behavior of the software which can be consulted for the formulation of the situation rules. With ROOM the behavior of the actors were specified by state-transition diagrams, which permit only predetermined combinations of the messages. A state-transition diagram can be expressed with the help of interval arithmetic, if the following points are observed:

- *the trigger condition* for a transition, i.e. the event which caused the state
- *the modification (actions)*, which are caused by a state
- *the history* of the preceding states.

The white-box modeling step consists of the transformation of the state-transition diagram into situation rules. After considering these situation rules, only the situations which describe the behavior of the actor remains. Similarly to the modeling of technical components, groups of situations can be commentated and classified. We can interpret these group of situations as scenarios for the suitable states of the ROOM-Chart.

[2] PI: Piping and Instrument Diagram

The design of the system model orients itself on the ROOM structure model. Due to the determined message channels, only certain messages can be exchanged between the actors and their environment. Similarly to the technical process, this fact is described with system equations. Subsequently, the calculation of the qualitative software model takes place with the help of a program. 50 of 4096 possible situations are suitable for the qualitative description of the controller software.

3.4 Qualitative Modeling of Human Task

In the presented example an operator has different possibilities to intervene into the process: He can open the door of the tank; start, terminate and abort the process and set new values for the filling level. Additionally he receives information from the software about the process status.

How do we model human behavior with SQMA? We consider the human as black box and we are only interested on his possibilities to intervene with the system. His access into the system is also represented as terminals of the component "human". The quantities of the terminals are described with interval variables, too. For example the qualitative variable for the new filling level contains all possible values from negative to positive infinitely. This means that humans can set any value. Actions of a humans can easily be expressed in a qualitative way (e.g. open the tank door, shut it, or do nothing).

In contrast to the modeling of the other part of the system, a black box modeling of the human is sufficient. This involves the modeling of unreasonable control interventions and logical intervention at the same time. For the safety analysis we suppose that a human can be e.g. in panic and don't act in the correct way. For the example "homogenizing plant" the qualitative model "operator" has 64 situations of different control interventions.

3.5 The Analysis of the System "homogenizing plant"

For the analysis, the models have to be joined together. To couple the model of the technical process and the software, additionally, the necessary actuators and sensors can be modeled in the qualitative way. Software information is converted into physical dimensions, or vice versa. A qualitative modeling of the field peripheries additionally enable the possibility of analyzing the system behavior with defective actuators or sensors.

Combinatorial 224000 situations are to be expected for the whole system. The specified system structure reduced the number of possible situations to 49. In the following situation table (Table 2), we can see an extract of the result. The first row contains the components of a special system part. For example the computer has calculated, that a situations can happen in which the inlet valve is closed, the tank is empty and the drain valve is closed. An adequate state of the controller software is that the actor plant controller is initialized and the actor plant dialog manager is ready, because human has started the system. By evaluating the attributes the computer declares this situations as intended.

As mentioned in chapter 2 the analysis is computer aided. A program sorts the system situations into intended (normal), not-intended and dangerous operations.

Table 2: An extract from the situation table of the homogenizing plant.

technical process			**controller software**		**human**	**status**
inlet valve	*tank*	*drain valve*	*plant controller*	*plant dialog manager*	*human task*	
...	...	...	...	...	...	...
closed	empty mixer off door closed	closed	initialization	ready	start system	intended
opened	contains fluid mixer on door closed	closed	filling/mixing	inputs blocked	nothing	intended
closed	full mixer off door opened	closed	initialization	ready	open door	dangerous
...	...	...	...	...	...	...

We can compare the system situations with a snapshot of the controlled system. In our example we have 36 situations, which are scenarios for the normal operation mode of the homogenizing plant. But for the safety analysis the other kind of situations are more interesting. With the help of the qualitative model the computer has detected 6 dangerous situations of the system. These situations describe the following scenario:

Liquid flows out through the door of the tank. With the developed software it can occur that liquid flows out from the open door of the tank. Depending upon the type and amount of the liquid the consequences can be differently serious. Nevertheless these system situations should never occur during the normal operation mode. Here, the qualitative model helps us to detect a design error of the controller software. These dangerous situations can occur only if the actor "plant controller" is in the state "initialization". If an analyst considers these situations, he will soon notice the causes for this failure: the process conditions at the beginning of the process is not checked by the software during its initialization. If e.g. in a preceded operation of the system the process was aborted by a user, liquid can still be in the tank. When renewed starting the system the door is unlocked again and the users has the possibility to open it (like 6 situations clearly show) or not.

Therefore the developers of the software are forced to change the software design. The state transition diagram of the actor "plant controller" is not sufficiently specified. A possible solution is to extend the state transition diagram by a state "checking process status". This new state has the function to lock the door first and only change into the state "initialization" if the tank does not contain any liquid. In the other case, it notice the user immediately and change into the state "draining".

When the discussed modifications of the software design are made, we can model the new design with SQMA once more and integrate it into the whole model for the system. After the renewed analysis of the model, the situations we discussed above shouldn't exist. But if so there must be new failures in the design and we have to think about it again.

4 Summary and Outlook

The presented example shows, how computer aided safety analyses can be realized with the help of the qualitative modeling method SQMA. The intended operation modes can be checked under any conditions, moreover the computer does announce not-intended and dangerous situations of the system to the user. This helps the analyst to judge the system safety. The result is reproducible at any time. Thereby, the quality of the safety analysis can rise noticeably.

The example demonstrates clearly how complex the interaction between the system parts can turn out, even for this small example. For larger systems it is almost impossible to discover or analyze several failures at the same time by conventional brainstorming.

Like other modeling techniques the quality of the result falls and rises with the underlying model. The component-oriented view of SQMA creates outstanding prerequisites for support by an existing and checked model-component library. Similarly to conventional CAD-Systems, models can be created on a graphic level comfortably.

5 Literature

[1] Leveson, Nancy G.: *SAFEWARE - System Saftey and Computers*, Addison-Wesley 1995

[2] S.Mohr & S.Montenegro, *Analysen der Anlagen-Sicherheit und –Gefahren*, GMD-First, http://www.first-gmd.de/~sergio

[3] Kuipers, Benjamin; *Qualitative Reasoning: Modeling and Simulation with Incomplete Knowledge*, Automatica, Vol. 25-4, 571ff, 1989.

[4] Xaver Laufenberg, *Ein modellbasiertes qualitatives Verfahren für die Gefahrenanalyse*, Dissertation, Institut für Automatisierungs- und Softwaretechnik, IAS, Universität Stuttgart 1997

[5] E Huber, G. Burgbacher, U. Biegert, W. Billmann, *Qualitative Systemanalyse und Computerunterstützte Gefahrenidentifikation*, Wiley-VCH, Chemie-Ingenieur-Technik, 7 / 97

[6] Biegert, Uwe, *Sicherheitsanalyse für Automatisierungssysteme*, 7. Kolloquium "Software-Entwicklung", Technische Akademie Esslingen (TAE), September 1997

[7] R. Lauber, P. Göhner, *Prozeßautomatisierung*, Band 1 und 2, 3.Auflage, Springer-Verlag Berlin Heidelberg New York 1999

[8] Selic, G. Gullekson, P. Ward, Real - Timer Object Oriented Modelling, John Wiley & Sons Inc., New York