# Identification of risks stemming from new communication technologies

**Lessis, Vasileios; Taylor, J.R.; Kozin, Igor**

*Publication date:*
2016

Enlarged Halden Programme Group Meeting in Sanderfjord, Norway, 8-13 May 2016. Paper code: MTO-4.2

# Identification of risks stemming from new communication technologies

*V. Lessis, J.R. Taylor and I. Kozine*
*Technical University of Denmark*
*+45 4525 4548, igko@dtu.dk*

## *Abstract*

*Advanced distributed communication technologies play an important role today in the control and maintenance of safety-critical systems. However, the excessively optimistic reliance on the new technology without recognizing the threats against its successful functioning, being able to maintain barriers or/and eliminate or reduce the risks may result in impairments compromising the opportunities. At the current state of knowledge it is even unclear whether we can develop trustful causal paths between hazards of different natures and their consequences. Hazard identification and risk analysis have proved to be effective tools in developing more reliable and robust systems. As technology is developing fast though, a new need for an effective hazard identification methodology has emerged. To enhance the predictive performance of hazard identification in advanced distributed communication systems, we have envisioned and currently developing a multilevel-multidimensional HAZOP methodology. The methodology introduces a new creative thinking stimulation model to substitute the conventional guideword-based approaches that is based on a multiple level and dimension exploration of the system under consideration. This paper describes the work carried out to extend the standard HAZOP approaches to suit the analysis of new communication technologies based on a simple corrective maintenance scenario taking place at a Nuclear Power Plant.*

## 1. Introduction

New communication technologies play an important role today in the control and maintenance of safety-critical systems. In the years to come, reliance on them will without any doubt grow, as they suggest benefits for operability, maintainability and efficiency of the systems up to the full automation of some functions which seemed infeasible some years ago. In particular, the new communication technology makes it possible to organize outage work process at NPPs and suggests impressive opportunities given by distributed computer technology. It is considered a logical next step towards enhancing team performance and the outage management process as a whole. However, excessively optimistic reliance on the new technology without recognizing the threats against its successful functioning can result in problems and delays in achieving the full benefits. Any new technology comes with new risks; and performing a risk analysis is necessary to introduce a more balanced view that pays attention to the threats inherent in the process.

In advanced distributed information systems, risks can stem from hardware and software failures, unintentional human errors, intentional manipulation, lack of situational awareness in some specific scenarios, malicious acts, misinterpretation, etc. At the current state of knowledge it is even unclear whether we can develop a full range of valid causal paths between hazards of different natures and their consequences.

A Control Systems Hazard and Operability (CHAZOP) analysis is a systematic procedure to identify potential hazards and/or operability problems in control and computer systems. However a standardized CHAZOP analysis does not exist. Though, there are a number of distinctive CHAZOP procedures focusing on identifying hazards in different subsystems and layers of information

systems, interfaces between them or/and different functional units. These can be programmable electronics, software, hardware inclusive communication devices and channels, human-machine interfaces and humans. Humans at the same time can perform very different functions at the systems and, as a consequence, can be viewed as different causes of risks. Some of the CHAZOPs can integrate few system levels and attempt to cover a wide range of possible deviations from the normal. However, while being operational and practical tools, none of them has so far been validated in order to be considered as a good engineering practice. Besides this, our experience in hazard identification and insights gained from an extensive review of the current methods make us believe that there is room for significant improvements in the predictability of hazards.

To enhance the predictive performance of hazard identification in advanced distributed communication systems we have envisioned and currently developing a multilevel - multidimensional HAZOP methodology; the MLMD HAZOP.

The diagrammatic causal representation of the system is made through the use of functional modelling methodologies which have a number of advantages over the use of other currently used representations as a basis for the analysis.

Besides this and aiming at the completeness and repeatability of the analysis, we introduce a new creative thinking approach, to substitute the conventional guideword-based approaches. The approach is based on a multiple level and dimension exploration of the system in place.

To demonstrate the approach, at this early stage of the project, we analyse a “toy” system developed through the initial design stages. The system chosen was one which follows the ethos in the Halden project “Outage Control Centres – Tools for Improved Team Performance” [1]. However, the actual “toy” case chosen is simpler and does not involve the complex problem of outage management but instead focuses on simple corrective maintenance.

## 2. State-of-the-art

Development and emergence of specialized hazard identification analyses focusing on communication technologies can be traced back to the late 1980-ies – early 1990-ies. In 1987 [2] HSE presented guidelines and safety principles concerning the use of programmable electronic systems (PES) in safety-critical domains. Following the clear need for a systematic hazard identification approach for PES Andow [3] proposed a 2-stage Computer Hazard and Operability (CHAZOP) analysis. CHAZOP for PES was built upon the HAZOP procedure which was well-established by that time having been developed for analysing chemical process hazards [4]. The first stage of CHAZOP was a preliminary CHAZOP addressing system architecture, safety-related functions and consequences of PES failures, site power and utility failures. The second stage was a full-scale CHAZOP analysing the environment in which the computer system works, identifying deviations in input/output signals, considering all possible control schemes and protective systems. The use of an updated list of guidewords was in the core of CHAZOP.

In 1992 Earthy published general guidelines for the use of HAZOP studies for software safety assessment [5]. As HAZOP studies for chemical and process plants are based on P&I diagrams, which cannot be used for the software analysis, the guidelines stress the importance of having proper basis for the safety assessment. As input to conducting a HAZOP study the guidelines suggest using task analysis models in the form of process flowcharts; while hardware (processors,

storage devices, input/output peripherals) should be presented by schematic network charts. Data Flow diagrams and transaction networks should also be used to define tasks in the system.

As further development, Burns and Pitblado [6] suggest a modification of the Failure Mode, Effect and Criticality Analysis (FMECA) with the use of HAZOP-type guidewords to identify hazards in PES. They propose guidewords in combination with variables such as “signal”, “information” and “action”. As input to the analysis, they name cause and effect charts and three types of diagrams: ladder diagrams, logic diagrams and vendor diagrams, with the logic diagram found to be the most convenient. The significant step forward in this paper was human factors analysis as part of hazard identification.

New sets of guidewords, deviations and parameters were proposed for the HAZOP analyses of electronic devices and software tools by Chudleigh in [7], [8]. As the natural input to the hazard analysis of electronic devices, block and circuit diagrams should be used; while the use of the “Software with Pictures” tool was advocated to create dataflow representation for the software.

Yet one more 3-stage methodology of hazard identification was developed by Mc Dermid and Pumfrey [9] to aid software design. Stage 1 addressed the software system representation that is made with the use of MASCOT [10]. In MASCOT components are represented by activities, Intercommunication Data areas and external devices that are all connected by information flows. Once the data type and path protocol of each information flow has been established (in the MASCOT diagram), at Stage 2 guidewords are defined by considering the interpretation of each failure class in the context of every combination of a data type and a path protocol. All failures are grouped into three classes as suggested by Bondavalli and Simoncini [11]: (1) service provision (omission, commission), (2) service timing (early, late), and (3) service value (coarse incorrect, subtle incorrect). At Stage 3 failure modes (deviations) are formulated following the recommended guidewords as well as causes and effects.

Fenelon and Hebbron [12] summarize the experience gained by that time on applying CHAZOPs and come up with some general recommendations. The first recommendation concerns system representation, the importance of which is that it controls the effectiveness of the CHAZOP. Representation should be able to show the interaction between software and environment, as software in isolation is not harmful. It is noted that top-down hierarchically decomposed models to represent the software are problematic because they require the analyst to anticipate the required division of functionality; which may lead to arbitrary decision being taken and, as a consequence, hazard visibility may be impaired. The second recommendation says that in order for a set of notations, methods and tools to be truly well-integrated, three levels of integration must be achieved. (1) Operational integration: a set of tools must be created which can interoperate with each other in a synergistic fashion allowing information from diverse sources to be assembled into a coherent whole. (2) Methodological integration: the methods implemented by the software tools must be fundamentally compatible and there must be meaningful procedures for translating results obtained from one notation into raw data for another. (3) Semantic integration: the models underlying the notations must be semantically consistent so that data from one part of a diverse set of notations retains its meaning when exported to some other notation. Finally, the guidewords should depend on the level of abstraction, to generate a meaningful deviation.

Kletz in [13] suggests a hazard identification method of computer controlled systems taking account of human error. The study starts with an “incident analysis” that derives causes and questions about past hazardous incidents. This is followed by a model of tasks and components that map questions and incidents with tasks. Each task is scrutinized against 5 considerations,

each of which is associated with a set of attributes/guidewords. He proposes a new graphical technique, called “Event Time Diagram” for viewing and decomposing tasks. In general, the method appears rather complicated to use.

Based on a literature review of hazard identification approaches Schubach [14] describes a modified computer hazard and operability study procedure. It is particularly stressed that HAZOP had been so successful because information about analyzed systems is provided in the appropriate form (P&IDs). Failure to create a concrete base documentation leads to poor analysis. A hazard analysis of computer controlled industrial processes should be based on P&IDs augmented with the necessary PES details. Finally, guidance to conduct the CHAZOP is provided as an 8-step activity procedure.

A HAZOP focusing on the identification of operator errors in interactive systems leading to hazards is described in [15]. The methodology is, first, to formulate and represent operational scenarios and define user goals for them. Second, for each user goal determine possible action sequences (tasks). Third, examine tasks, using HAZOP-like guidewords. The core of the methodology is a task analysis (objects, actions, goals, tasks) of a safety-critical system that is supported by the use of the Sum language. The model produced by the task analysis shows all operations in the scenario and is used as a base for the HAZOP-like analysis. For each operation, guidewords are used to determine failure operations that in the following step are substituted with the normal operations in the model and the outcomes are observed.

Hansen, Wells and Maier [16] describe HAZOP for UML design diagrams and is very much in line with the objectives of this paper with multiple aspects of the design and multiple HAZOP check lists. The approach uses guide words only, and as a result leaves the analysis approach somewhat incomplete, at least as far as the present objectives are concerned.

Perhaps the most comprehensive approach to a hazard analysis of computer systems is described in Process Automation Handbook [17]. This is again a 2-phase CHAZOP analysis. Phase 1 consists in preliminary CHAZOP integrated with a full CHAZOP. The preliminary CHAZOP includes strategic decisions about segregation policy, system layout, and the handling of gross failures. The results of the analysis are incorporated in the user requirements specification (USR) for the control system. The full CHAZOP goes first through each I/O signal to establish whether it is used for any safety related function. Then it identifies all communications channels used by any such signal. The following step scrutinizes the channels against deviations from intended functioning by the use of proper guidewords. In doing so, particular attention should be paid to the segregation and redundancy requirements. Finally, the results should be incorporated in the hardware aspects of the detailed functional specification (DFS). Phase 2 is basically a Control and Operability (COOP) study for the application software and human factors. In essence, a COOP study is used to check whether the design of the application software takes properly into account all conceivable and relevant human factors.

Table 1 summarizes the key characteristics of the reviewed CHAZOP methodologies.

## 3. Motivation for the present research

Development of many modern operations management systems have failed, been delayed, or have led to major accidents [18]. Hazard identification and risk analysis have proved to be effective tools in developing more reliable and robust systems. As technology is developing fast, especially

with the introduction of distributed communication systems, a new need for an effective hazard identification methodology has emerged.

| # | Author | Year | Input | Software | Hardware | Human Interaction | Creative Thinking stimulated with |
|---|---|---|---|---|---|---|---|
| 1 | Andow | 1991 | - Computer System / Environment<br>- I/O signals<br>- Control Schemes | | ✓ | | - Guidewords<br>- Guiding Questions |
| 2 | Burns, Pitblado | 1993 | - Logic Diagrams<br>- VDU Displays<br>- Control room layouts<br>- Access times<br>- System Interfaces<br>- Job role & task definitions | | ✓ | ✓ | - Guidewords<br>- System Parameters |
| 3 | Cludleigh | 1993 | - Block & circuit diagrams<br>- Data flow diagrams | ✓ | ✓ | | - Guidewords<br>- Deviations<br>- System Parameters |
| 4 | McDermid and Pumfrey | 1994 | - MASCOT designs | ✓ | | | - Dynamically generated failure modes / deviations |
| 5 | Kletz | 1995 | - Incident Analysis<br>- Event Time Diagram | ✓ | ✓ | ✓ | - Guiding Questions |
| 6 | Schubach | 1997 | - P&IDs augmented with PES details | | ✓ | ✓ | - Guiding Questions |
| 7 | Hussey | 2000 | - Task Analysis / Scenarios modelled with Sum Language | | ✓ | ✓ | - Guidewords<br>- Deviation Modelling |
| 8 | Hansen, Wells and Maier | 2004 | - UML Diagrams | ✓ | ✓ | ✓ | - Guidewords |
| 9 | Love | 2010 | - URS and DFS, P&I diagrams, Channel / Loop Diagrams, Software designs, Control System Architecture / Hardware, I/O Signals, Function Block Diagrams, Sequential Function Charts, Sequence Flow Diagrams | ✓ | ✓ | ✓ | - Guidewords |

*Table 1. Comparative review of CHAZOP analysis*

A number of CHAZOP variations have been suggested since the early 90's. However, due to a number of reasons they are regarded as insufficient to address potential risks found in distributed communication systems. Some of the most important issues that current approaches face include:

- They generally focus on specific aspects of the system, such as the software or hardware and not the system as a whole; interactions are disregarded or not treated sufficiently.
- The base documentation proposed is inappropriate or insufficient; failure to create a concrete base documentation inevitably leads to poor analysis.

- Several current check lists/methods published are similar but not consistent from publication to publication. The check lists tend to mix deviations appropriate to different technologies.
- Many of the terms used are not orthogonal i.e. some disturbance types are covered by several checklist items.
- Some sets of terms do not seem to be logically complete, at least with respect to our objective.

The purpose of the current project is to develop and validate a methodology, able to address these issues and more specifically investigate in-depth the notions of completeness, through a multilevel - multidimensional analysis.

## 4. Methodology

HAZOP analysis, was selected as the base of our methodology as it has proved to be extremely successful in the process industries and its effectiveness and completeness has been well documented [19]. More specifically, the CHAZOP variant was chosen as a basis since it is directly relevant for control systems and process management. The success of HAZOP is dependent on using guide words and a related list of deviations to provide a sound basis for identifying starting points for causes and consequences of system failures, upsets and accidents.

### 4.1 System Representation

Conventional HAZOP uses piping and instrumentation diagrams (P&IDs) as the base documentation to represent the system under investigation. While this decision might seem trivial now, it plays a significant role in the success of the methodology. A documentation lacking to represent aspects of the system can and will most certainly lead to an incomplete hazard identification analysis, which can have disastrous aftermaths. Existing CHAZOP methodologies use a number of different base documentations, depending on the aspects of the system that they focus on. Nevertheless, none of them uses one with the ability to fully represent a distributed communication system.

The base documentation we propose in this paper is summarized in Table 2. Further explanation is provided in the following sections.

| **Documentation for System Representation** | |
|---|---|
| 1) Computer Network Diagram | Overall System Architecture / Hardware |
| 2) UML Activity Diagram | What activities are undertaken and by whom - Synchronization and Coordination |
| 3) IDEF0 diagram | How activities are undertaken - Support system |

*Table 2. Proposed documentation for system representation*

#### 4.1.1 **Computer Network Diagram** *(Appendices Figure 1)*

A computer network diagram in its classical form is a schematic depiction of all the nodes and connections found in a telecommunications network. In other words, it represents all the hardware used in a system including the transmitters, receivers and channels that information uses to travel

throughout the network. The computer network diagram is the perfect match to give an overview of the system architecture and to be used as a base for identifying potential hazards related to the hardware in place.

#### 4.1.2 UML Activity Diagram *(Appendices Figure 2)*

The second document proposed, is the UML Activity Diagram. Activity diagrams are graphical representations of the activities, actions and workflows that are undertaken in a system. Essentially they are like flowcharts, however they are also equipped with means for expressing *concurrency* (join and split symbols); concurrency is an important aspect to take into account in hazard identification. The UML activity diagram is used to represent *what* activities are undertaken and by *whom*, including aspects of synchronization and coordination.

#### 4.1.3 IDEF0 Diagram *(Appendices Figure 3)*

IDEF0 is a functional modelling methodology, widely used in describing both automated and non-automated systems. The basic philosophy behind it is that every function or activity is represented by a box. *Inputs* are represented by arrows entering the left side of the activity box, while *outputs* by arrows exiting its right side. Moreover, the *mechanisms* used to accomplish the activity are represented by arrows entering the bottom of the box, while the *controls* that direct each activity by arrows entering the top. While the UML activity diagram is used to show the workflows into the system, the IDEF0 diagram is used to show the data flow, system control and the mechanics behind each activity; the support system. In other words, it is used to represent *how* each activity is undertaken and how the actors interact with the hardware, software or/and each other.

The base documentation proposed was not selected arbitrary and it should be used combined in order to sufficiently represent a distributed communication system and all the interactions that come with it. Another reason behind our proposal is the fact that the above-mentioned representations become used in the industry and engineers become more and more familiar with them.

### 4.2 Study Team

One of the main features of the HAZOP analysis family is that they are essentially group activities. Ideally the team performing the analysis should have members representing a range of expertise and disciplines. The premise underlying this is that no one can have a range of knowledge which can cover all aspects of failure and error. This premise is certainly true for distributed information systems, where knowledge of the technology, the programming and the application are important.

Through the years, HAZOP has achieved an important status in some industries, to the extent of it becoming a legal requirement in some areas. The reasons for this are that a multi person, multi discipline view of an system is obtained, that the view is constrained by the methodology to be a consensus, and that the methodology is aimed at proposing methods to improve safety.

The methodology proposed in this paper suggests as a bare minimum for the study team to consist of:

- A software engineer
- A network architect/engineer
- A maintenance engineer (since the project studied concerns maintenance, more generally a problem domain expert)

- A system operator

### 4.3 Multilevel – Multidimensional Analysis

Existing CHAZOP variations use a list of guidewords and/or relative system parameters in order to stimulate creative thinking for identifying potential hazards and operability problems. However, we already mentioned some of the issues that these approaches face.

Hulin and Tschachtli [20] identified in particular four methodological and technical insufficiencies that guideword-based approaches face;

- *Incompleteness*; proposed guidewords cannot identify all potential deviations.
- *Nonsensicality*; some guidewords are not straightforward, thus they cannot be logically associated with any deviation.
- *Redundancy*; a number of guidewords can lead to the same deviation.
- *Ambiguity*; one guideword can lead to a number of different deviations.

The methodology we propose aims to directly deal with the issues of incompleteness and nonsensicality and create a logical framework where, while redundancy and ambiguity naturally exist, they can be easily handled and used to one's advantage.

The multilevel - multidimensional methodology divides the given system into levels and then uses a multidimensional guiding philosophy for identifying potential hazards. At the current state of the toy system design, the methodology divides the system in two levels;

- Hardware Level, including all the transmitters, receivers, channels and I/O signals.
- Human Interaction Level between software, hardware and operators.

For each level of the design, an appropriate deviation check list is developed, by taking a generalized multi-dimensional functional failure check list and interpreting this in relation to the actual design to be analysed. The master check list was obtained by dimensional analysis of the aspects of functioning and functional failure [21].

Starting with the hardware level, the study team uses as input the *computer network diagram*. In order to stimulate creative thinking towards hazardous deviations, a two-dimensional guiding framework is used, Table 3. Since we deal with hardware components at this level, the first dimension of the framework separates the components in "continuous" working ones and "on demand" working ones. A third row is also used to represent both types of components. The second dimension classifies deviations according to three different criteria: "working state" of the component, "time" and "component interaction" with the rest of the system.

By looking at the two dimensional matrix, it is fairly straightforward to populate it with potential *guiding deviations.* This list is a subset of the more extensive dimensional analysis of deviations, being selected for application for distributed control and management systems.

For the human interaction level, the study team uses as input both the UML activity diagram and the IDEF0 functional block diagram. Creative thinking is also stimulated here through the use of a multidimensional guiding framework, Table 4, although the dimensions are selected to match the human interaction with the software, hardware and other humans. The first dimension separates the human interaction in regard to the "data" that is being transferred, the "time" of the interaction and finally the "place" that the interaction occurs. The second dimension considers all the different possible interactions that take place in a distributed communication system. More specifically, we have interactions between human - machine and human - human. Human - machine interactions

can be further separated into “input-to-machine” interactions and “output-from-machine”. Human - human interactions are represented in the table with the “interaction with other human” column.

| | Deviations Related to | | |
|---|---|---|---|
| **Intended to work** | **Working State** | **Time** | **Component Interaction** |
| **Continuous** | - Working<br>- Partially working<br>- Not working (fails while working) | - Interrupted<br>- Intermittent | **Not Applicable** |
| **On Demand** | - Works on demand<br>- Partially works on demand<br>- Fails on demand | - Premature / Early (Too early starting)<br>- Delayed / Late (Too late starting)<br>- Too fast<br>- Too slow | - Duplication<br>- Omission<br>- Sequence error |
| **Both** | - Wrong function | - Prolonged / Extended (too long duration)<br>- Curtailed / Partial (too short duration) | - Synchronization failure<br>- Wrong Version<br>- No/Limited Access<br>- Interference<br>- Overload |

*Table 3. Hardware level – Multidimensional guidance matrix*

| | Deviations Related to | | |
|---|---|---|---|
| **In Regard to** | **Input - to - machine** | **Output - from - machine** | **Interaction with other human** |
| **Data** | - No input<br>- More input<br>- Less input / Incomplete<br>- Wrong input<br>- No/ Limited access | - No/Loss output<br>- Overload (too much output)<br>- Less output / Incomplete<br>- Wrong output<br>- Ambiguous output | - Confusion<br>- Conflict |
| **Time** | - Early input<br>- Late input<br>- Input Before<br>- Input After<br>- Synchronization error | - Early output<br>- Late output<br>- Output Before<br>- Output After | **Not Applicable** |
| **Place** | - Inaccessible machine for input | - Inaccessible machine for output | - Inaccessible |

*Table 4. Human Interaction Level – Multidimensional guidance matrix*

The matrix is populated once again with possible deviations that can be used as a guidance from the study team.

While most methodologies propose the use of guidewords and checklists to identify potential deviations, the methodology presented in this paper recognizes that there are different design

domains, and that these will require different check lists, or at least different vocabulary, and will need to use a multilevel framework. The higher levels of analysis “inherit” the deviations and consequences from lower levels, and lower level consequences are used to supplement the check lists used at higher levels. Although the multilevel concept has been proposed before, the use of dimensions gives a sense of “logical completeness” and offers a systematic way to the study team to consider possible deviations. The complete steps of the methodology are presented in Figure 4 in the Appendices

## 5. Case study

In order to demonstrate the applicability of our approach and validate the methodology, a simple corrective maintenance scenario was developed. The scenario involves a Nuclear Power Plant and concerns design of a work order processing system in a routine corrective maintenance. It was ambitiously titled Distributed Maintenance Management System (DMMS). This allows us to draw on experience from earlier systems developed for management of integrity of safety critical systems. For simplicity this “toy” design does not include outage management, preventive maintenance, condition based maintenance or diagnostic support, and the safety management aspects are highly simplified.

The scenario involves the communication of four different actors, distributed across the NPP;

- *Operations Centre*; Responsible for all maintenance operations, monitoring and updating the work schedule.
- *Work Planning Centre*; Responsible for physically opening and closing work windows/orders, planning and re-planning.
- *Main Control Centre*; Responsible for overall plant safety, issues clearance orders, tag out papers etc.
- *Field Workers*; Receive and perform work orders.

Following the methodology, in order to represent the system, a computer network diagram, a UML activity diagram and an IDEF0 functional block diagram were developed (see Appendices Figure 1, Figure 2, and Figure 3 respectively).

## 6. Results

The analyses for the selected system were carried out by a team of three, one with risk analysis experience, one with IT experience and one with maintenance experience. This would be a bare minimum for a practical project, but it satisfies at least the requirement for a multi-disciplinary approach. Some results of the analysis are presented in the Appendices Figure 5.

The DMMS was analysed first at the communications and hardware level. This took six hours, identified 20 failure possibilities, no design errors, and 11 proposals for safety measures. The level 1 functional analysis was more separate, with the analysis spread over several days. It took a total of 22 man-hours, identified 3 design errors, and 44 proposals for safety measures.

At the outset of the functional analysis, the support systems were recorded on the IDEF0, in accordance with the IDEF notation standard. This allowed the results from the hardware level analysis to be cross-referenced to the higher level functions which they would affect. The low level disturbances could then be “inherited upwards” and provided additional causes for the higher level disturbances.

One of the most surprising results from the analysis was the early detection of a major design error in the initial concept, within the first hour of work. This would presumably have been found later in the design process, even in the absence of the HAZOP analysis, because of the simplicity of the design. However the example illustrates the way in which the performance of systems can be impacted by errors at the conceptual stage, especially in complex systems.

The most important result from any HAZOP is the documentation of existing safety measures, the proposals for new measures, and the allocation of responsibility for error remediation. Because of the stage of design it was not possible to distinguish between “existing” and “proposed”. The list could nevertheless go forward to later stages of design, for evaluation and direct incorporation. After evaluation, these could be consolidated into a “safety and reliability philosophy document” of the kind which is at present used in the field of process engineering.

One this that became obvious during the HAZOP was the importance of “design rules” which were being used particularly for selecting safeguards, but also a supplement to the deviation check-lists. These correspond to the design review rules which are actually used alongside the formal HAZOP in process engineering applications. An example of a design rule is:

Any function in a system can be interrupted, either because of hardware failure or due to a mismatch between the function preconditions and the actual state of the external world. For any function or functional sequence there should be a) a possibility to put the function “on hold”, and b) a recovery function.

## 7. Conclusion and Future Work

The project is at a very early stage, so that it is too early to draw general conclusions. The future development includes:

- Extension of the “toy” system to deeper levels in order to validate the principle of multilevel analysis.
- Further collection and documentation of design rules
- Extension of the methodology to the human machine interactions level (level 3 in our conjectured design process.
- Comparison of the results of the analysis with an existing database of information management system failures.
- Evaluation of the different design notations.
- Development of a guideline for selection and development of safeguards for information processing systems which will complement the hazard analysis methodology.
- We hope to apply the methodology to a significant design.

## 8. References


[1] P. L. D. Lars Hurlen, “HWR-1142 Outage Control Centers – Tools for Improved Team Performance,” in *OECD Halden Reactor Project*, 2014.

[2] Health Executive Safety, “Programmable Electronic Systems in Safety Related Applications: 1. An Introductory Guide 2. General Technical Guidelines,” *HMSO*, 1987.

[3] P. Andow, “GUIDANCE ON HAZOP PROCEDURES FOR COMPUTER - CONTROL PLANTS,” 1991.

[4] C. D. Swann and M. L. Preston, “Twenty-five years of HAZOPs,” *J. Loss Prev. Process Ind.*, vol. 8, no. 6, pp. 349–353, 1995.

[5] J. V. Earthy, “Hazard and operability studies as an approach to software safety assessment,” in *Hazard Analysis, IEE Colloquium on*, 1992, pp. 5/1 – 5/3.

[6] D. . Burns and R. . Pitblado, “A modified HAZOP methodology for Safety Critical System Assesment,” *Proc. Safety-Critical Syst. Symp.*, 1993.

[7] M. F. Chudleigh and J. R. Catmur, “Safety assessment of computer systems using HAZOP and audit techniques,” *Proc. SAFECOMP’92*, 1992.

[8] M. Chudleigh, “Hazard Analysis Using HAZOP : A Case Study,” in *SAFECOMP ’93*, 1993, pp. 99 – 108.

[9] J. A. McDermid and D. J. Pumfrey, “A development of hazard analysis to aid software design,” *Proc. COMPASS’94 - 1994 IEEE 9th Annu. Conf. Comput. Assur.*, pp. 17–25, 1994.

[10] JOINT IECCA & MUF COMMITTEE ON MASCOT, “The Official Handbook of Mascot,” p. 324, 1987.

[11] A. Bondavalli and L. Simoncini, “Failure classification with respect to detection,” *[1990] Proceedings. Second IEEE Work. Futur. Trends Distrib. Comput. Syst.*, pp. 47–53, 1990.

[12] P. Fenelon and B. Hebbron, “Applying HAZOP to software engineering models,” *Risk Manag. Crit. Prot. Syst. …*, 1994.

[13] T. Kletz, P. Chung, E. Broomfield, and C. Shen-Orr, *Computer Control and Human Error*. 1995.

[14] S. Schubach, “A modified computer hazard and operability study procedure,” *J. Loss Prev. Process Ind.*, vol. 10, no. 5–6, pp. 303–307, 1997.

[15] A. Hussey, “HAZOP analysis of formal models of safety-critical interactive systems,” *Comput. Safety, Reliab. Secur. Proc.*, vol. 1943, pp. 371–381, 2000.

[16] K. M. Hansen, L. Wells, and T. Maier, “HAZOP analysis of UML-based software architecture descriptions of safety-critical systems,” *Proc. NWUML*, pp. 1–23, 2004.

[17] J. Love, *Process Automation Handbook: a guide to theory and practice*. Springer-Verlag London, 2007.

[18] P. G. Neumann, *Computer Related Risks*. ACM Press/Addison-Wesley Publishing Co. New York, NY, USA, 1995.

[19] J. R. Taylor, “Forty Years of HAZOP - Lessons Learned,” *Loss Prev. Bull. I.Chem E*, 2012.

[20] B. Hulin, D. B. Ag, and R. Tschachtli, “Identifying Software Hazards with a Modified CHAZOP,” *PESARO 2011 First Int. Conf. Performance, Saf. Robustness Complex Syst. Appl.*, no. c, pp. 7–12, 2011.

[21] J. R. Taylor, “Design Error, From Hand Axes to Nuclear power and Artificial Intelligence,” *ITSA*, 2016.

## 9. Appendices

**Figure 1**: Computer Network Diagram – Corrective Maintenance Scenario

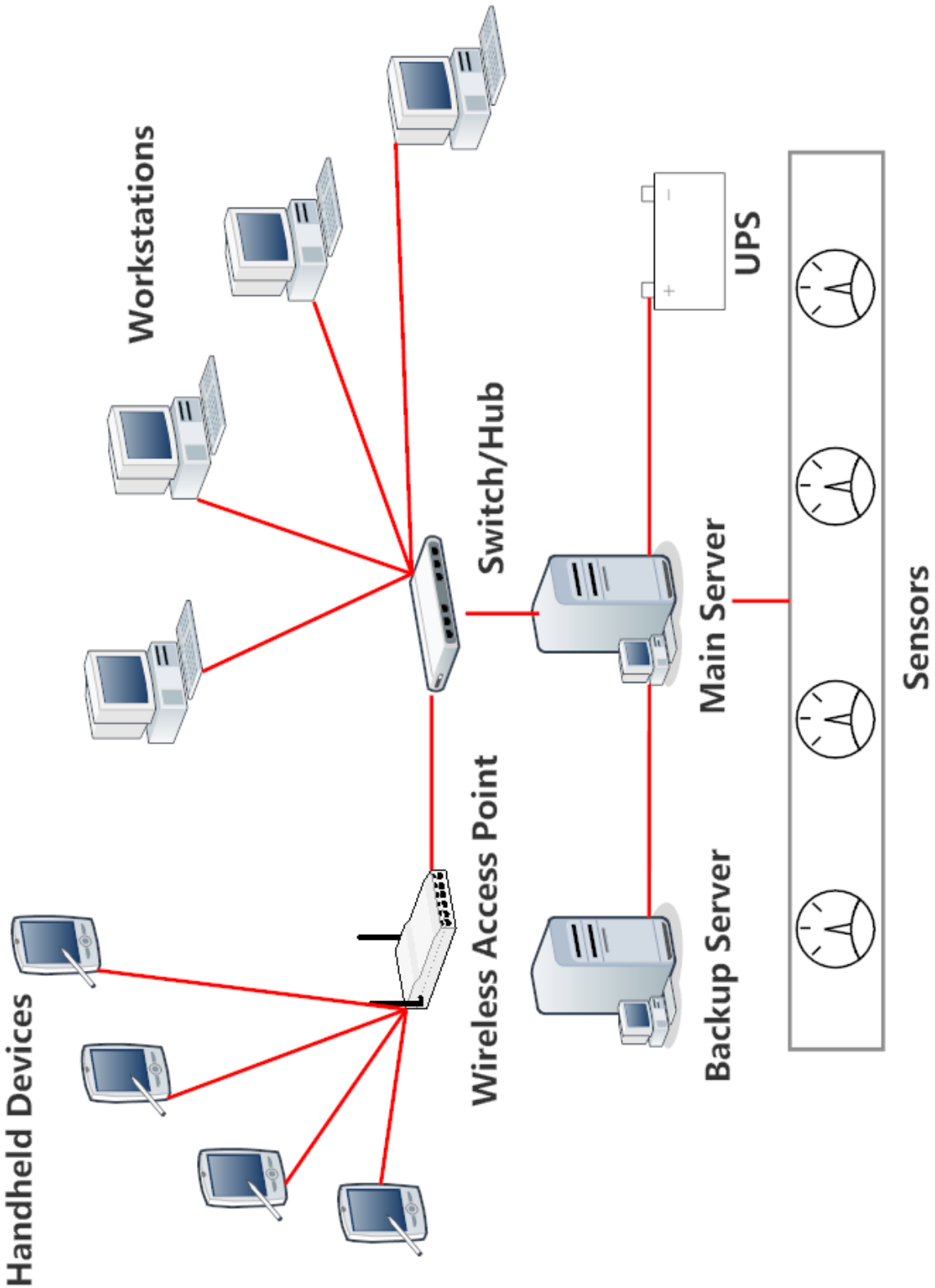

**Figure 2**: UML Activity Diagram - Corrective Maintenance Scenario

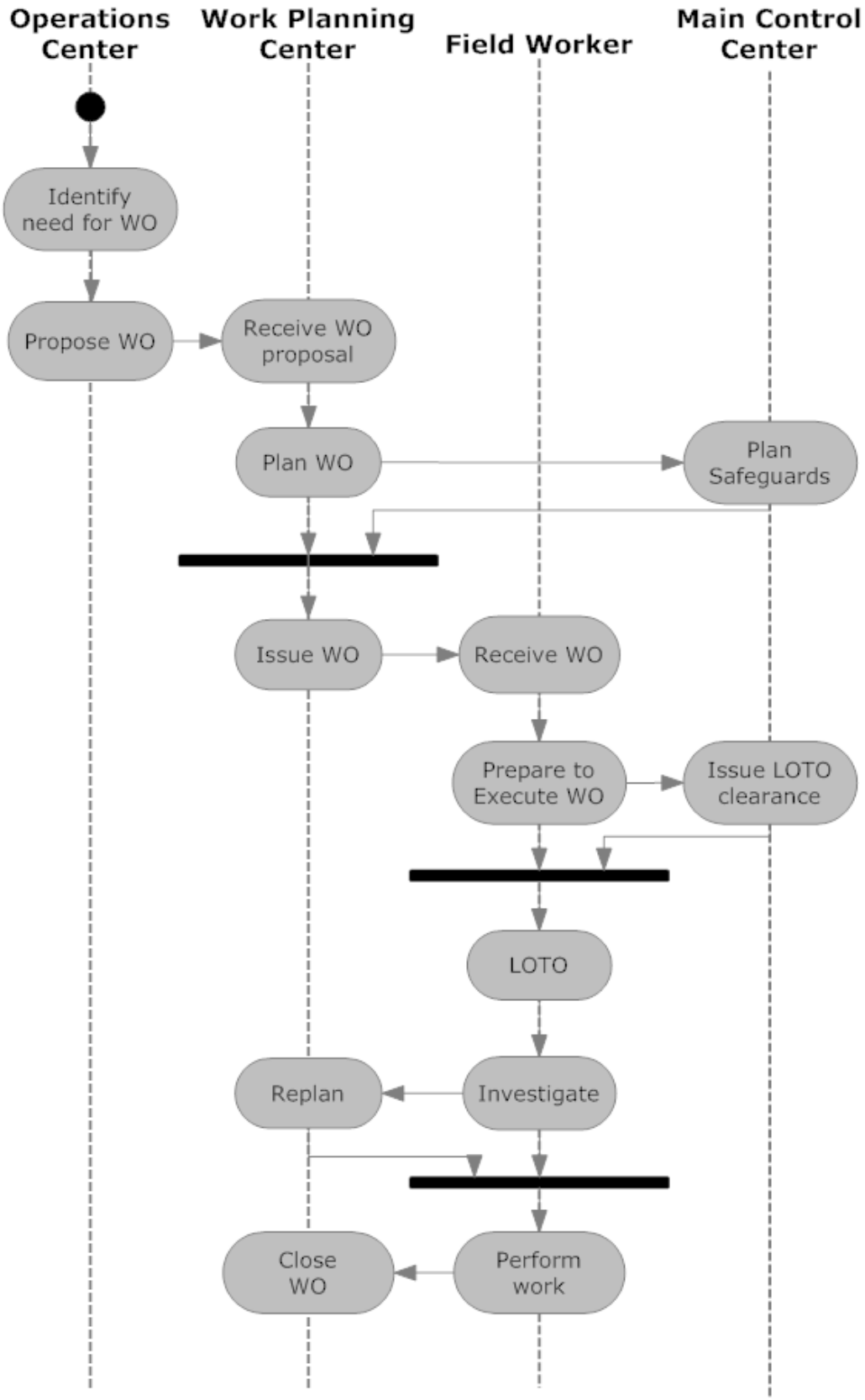

**Figure 3**: IDEF0 Diagram - Corrective Maintenance Scenario

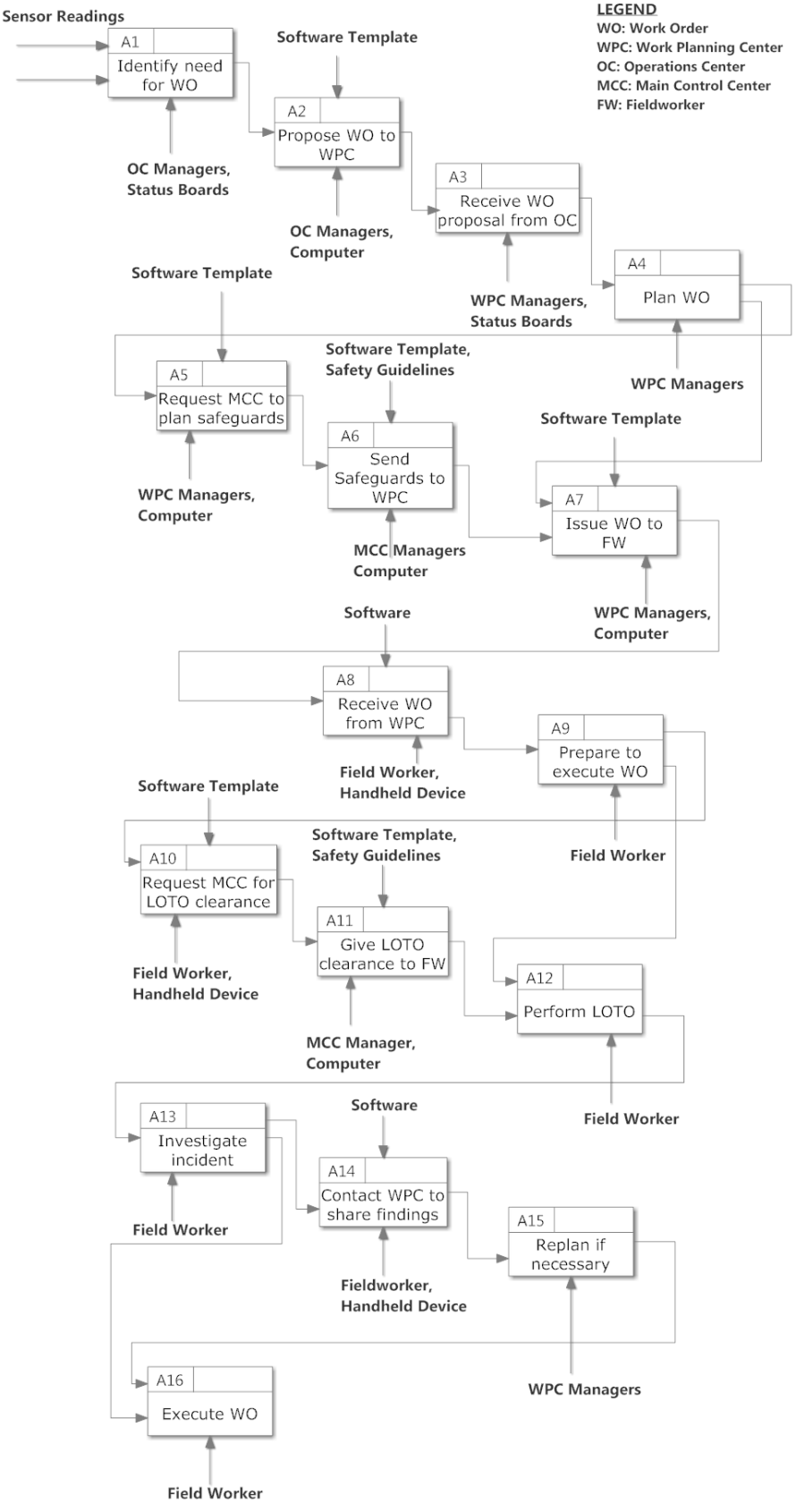

**Figure 4**: Multilevel - Multidimensional HAZOP Procedure

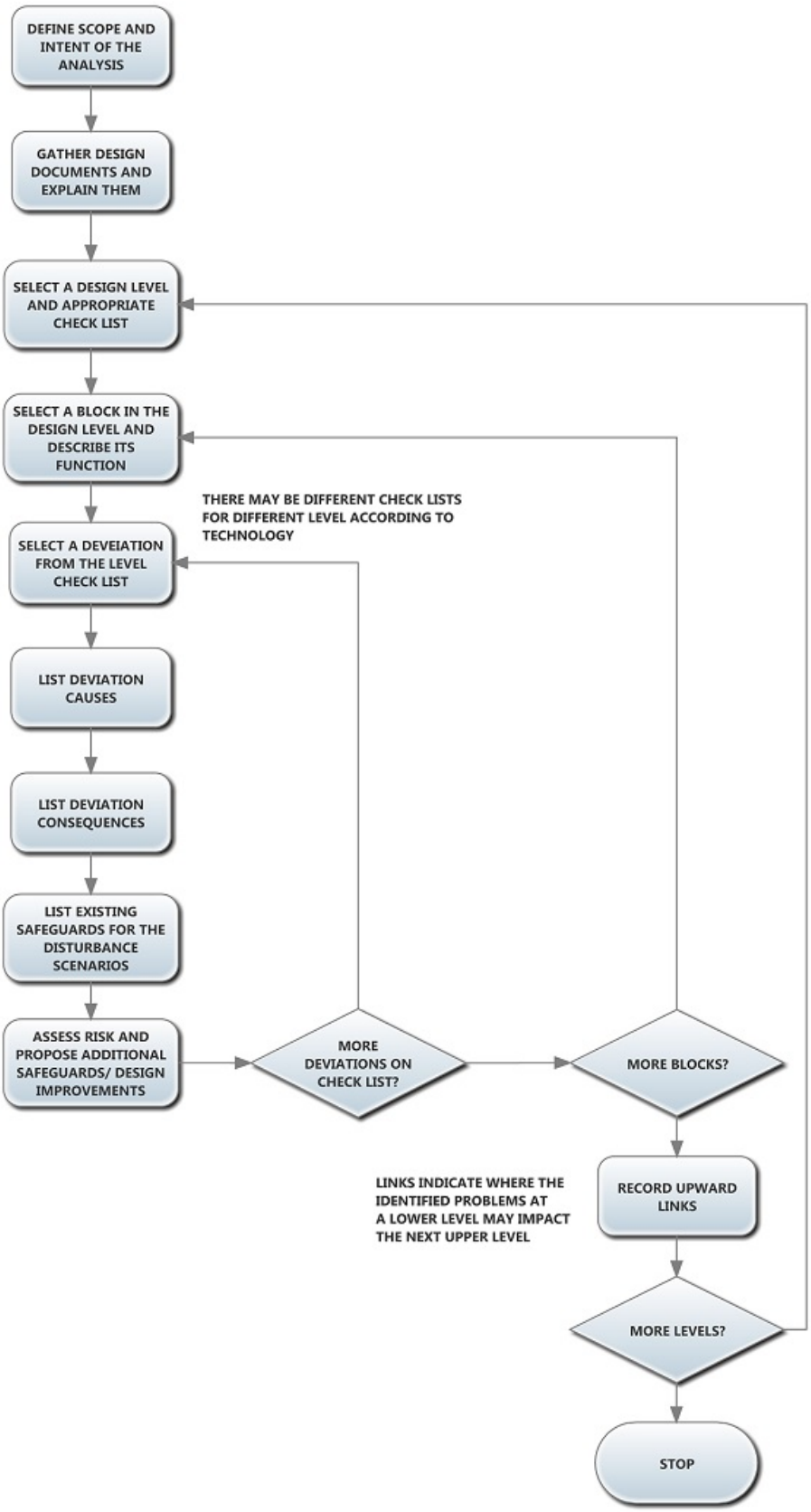

**Figure 5**: Multilevel – Multidimensional HAZOP Results (one-page extract from the report)

| Analysis | MLHAZOP for IDEF0 Level 1 Specification Interface Deviation Analysis | | | Participants: JRT | | Date: 25/02/2016 |
|---|---|---|---|---|---|---|
| Object | Deviation | Causes | Consequences | Existing safeguards | Impact from | Recommendations |
| **A1 Identify need for WO** | No input | Sensor malfunction<br>Alarm and status display malfunction function<br>Alarm and status<br>Window not displayed or covered.<br>Alarm and status display overload<br>Failures from level 4, process display and operator work station | Initially no work order created, delay in activating corrective work<br>-> Delay in corrective maintenance<br>--> Failure escalation<br>--> Consequence escalation<br>---> Possible accident, IP | Routine inspections<br>Secondary alarms<br>Emergency detection systems (fire, smoke, leak, gas, radiation)<br>Redundancy e.g. in the form of condition monitoring | Level 0, System input | Consider inputs from common cause failure analysis |
| | More Input | Too many problems arising, overload, operator cannot respond | As above, No input | As above, No input<br>Also:<br>Priority system on alarms | Level 0, System input | |
| | Less input | Alarm or display system failure | As above, No input | As above, No input | Level 0, System input | |
| | Incomplete input | As above, No input | As above, No input | | Level 0, System input | |
| | Wrong input | Spurious alarm, alarms fail on demand | Unnecessary maintenance work initiated. | Monitoring and process parameter displays supplementing alarms | Level 0, System input | Comment: Distinguishing spurious alarms from real ones is a typical first step in maintenance |