Proc. of The Third Intl. Conf. on Advances in Computing, Electronics and Communication - ACEC 2015

ISBN: 978-1-63248-064-4 doi: 10.15224/ 978-1-63248-064-4-18

Fault Effect Analysis based on a modelling Approach for Requirements, Functions and Components

Huiqiang Wang, Nasser Jazdi, Michael Weyrich

***Abstract*— At present most of fault diagnosis systems are dedicated to fault detection, fault elimination and fault effect analysis with specific diagnosis approaches. Even though fault effect based on these approaches concerns the affected components as well as functions. However, there is no analysis of the still available functions which could continue to operate despite the failure. Currently, there are few approaches working on the fault analysis based on the existing fault diagnosis system. To support a full fault diagnosis, this paper proposes a novel full fault effect analysis approach based on system models. Within a defect component in an automation system, available functions can be identified by the presented approach. The presented approach uses the results of the existing fault diagnosis system as an analysis basis, e.g. the fault ID and the fault location. The propagation of a fault is identified with the help of the requirement-function-component models, which are provided by the manufacturer. As a result, the presented approach is able to identify the available functions in case of the appearance of defect components. In this way, it enables an industrial automation system to continue its operation with restricted functionality despite a fault.**

Keywords— ***requirement model, functional model, component model, fault effect analysis, function degradation, requirement dependency***

I. Introduction

Automation systems can have a harsh impact on operations level in a company, if failures occur at the adverse time [1]. This can result in extreme financial loss, customer dissatisfaction and loss of productivity. In case of a problem, normally not all functions are affected, but still this often results in a downtime of the whole automation system, even though, a subset of functions could still be available. Hence, in this paper a novel methodology for the system operation is proposed, which determines available functions and defect components. In order to deal with the issues of fault defect analysis in automation systems, a remote problem management system is previously proposed in [2]. The main objective of this problem management system is to assure that the automation system continues to work during the occurrence of failures with decreased functionalities. The problem management system consists of a "passive management" part to cope with the known problems and an "active management" part to deal with unknown problems, i.e. those problem that take place for the first time (See figure 1).

Huiqiang Wang, Nasser Jazdi, Michael Weyrich
Institute of Industrial Automation and Software Engineering
University of Stuttgart
Germany

For passive management, the available functions, which are stored in the manufacturer's knowledge base, can be inquired by the remote problem management system directly. Afterwards, the information about the available functions are sent back to the automation system. The defect automation system is reconfigured with the available functions.

In the active management, unlike known problems, the problem management system must identify the effect of defect components in accordance to the fault diagnosis results [3]. Then the available functions are determined by the problem management system. These functions are stored in the manufacturer's knowledge base on one side and are sent to the industrial automation system on the other side. Then the automation system is reconfigured with the available functions.

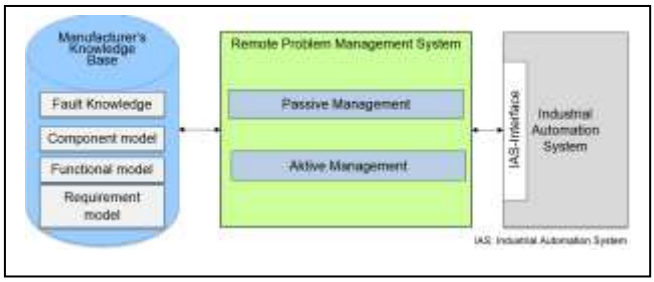

Figure 1. Schema of the remote problem management system [2]

In order to determine the impact of a problem, this paper propose a model based approach to analyze the problem propagation in automation systems. As a result, the available function can be determined and maintained. For this purpose, different models with different scopes are used for the analysis of automation systems. With the help of the component model, it is possible to recognize affected components by the defect components as well as subsystems. The functional model is helpful in detecting the affected functions, as well as not affected functions. Afterwards, the domain dependency between functions are verified by means of the requirement model, which defines all the non-functional requirements.

After the introduction, section 2 describes related works, the relationship between the fault diagnosis and the description of automation system models. Section 3 describes the concept of identifying the available functions based on automation system models. Section 4 deals with the prototypical realization of the proposed concept on an industrial coffee dispenser. Section 5 provides a brief conclusion and further works.

Proc. of The Third Intl. Conf. on Advances in Computing, Electronics and Communication - ACEC 2015

ISBN: 978-1-63248-064-4 doi: 10.15224/ 978-1-63248-064-4-18

II. Background and Theoretical Basis

A. *Related work*

In an automation system fault tree analysis (FTA) [4] and failure mode effect analysis (FMEA) [5] are the typical systematic techniques to identify fault effect. They are usually applied to analyze the fault and failure effect for the known faults and failures. Fault tree analysis is a systematic and stylized deductive process. The main applied fields are safety engineering and reliability engineering. The purpose of this approach is to identify the reason how the systems will be failed, to recognize the proper means to decrease risk and to confirm event rates of a system fault at a particular level, e.g. system level. However, this approach depends on the predefined events of faults in an automation system, which have been clarified in the development phase, e.g. condition-based fault tree analysis [6]. In [6]; the condition-based fault tree analysis starts with the known fault tree analysis. And the reliability values of a specific system are updated during the overall product life. A functional-failure identification and propagation framework is introduced in [7], which enables the developer to analyze the functional failures and their propagation paths and to determine what function are lost. However, this approach is only helpful to the conceptual design. It lacks the explicitly formulate a fault propagation model and the specific model of the automation system. In [8], a function-behavior-structure approach is proposed to represent an automation system in function, behavior and structure view. Regarding these models, the effect of a fault can be analyzed in the design phase. However, this approach cannot precisely describe the automation system. With regard to the abstract view, this approach considers no requirements to verify the availability of the functions. Towards the physical aspect, the components are not detailed described instead of only limited predefined configurations. It cannot perform a full fault effect analysis procedure and also obtain the available function in case of a fault.

Hence, there is no proper approach to analyze the fault effect during the operation phase, i.e. to obtain the available functions in case of a fault. In this paper, a novel fault effect analysis approach is proposed, which uses the models defined in the development phase, including requirement-function- component model.

B. *Relationship with the Fault Diagnosis*

Nowadays a fault diagnosis system is considered as a necessary method to deal with the problems and enhance the availability of an automation system. A large percentage of research concentrates on the fault diagnosis, such as the improvement of the diagnosis tools [9] and the improvement of the fault data processing [10]. However, most fault diagnosis system are used to support the maintenance service to remove known faults [10]. But they are short of the analysis of the still available functions. Meanwhile, there are few approaches working on the fault analysis based on the diagnosis results of the existing fault diagnosis system. Hence, this analysis approach is using the fault diagnosis results and makes a secondary analysis.

On the contrary, this approach aims at the analysis of the problem impact on the basis of the fault diagnosis results automatically. So that an unnecessary waste of resources can be avoided. Hence, this approach assumes that the fault diagnosis is already performed. As an indispensable part of the fault diagnosis result, the location of a problem has been confirmed in two views, a defect component as well as a defect subsystem, in which the responsible sensor is known. Firstly, a defect component, is easy to understand, such as, a defect valve, a defect sensor, or other components. For a subsystem, the necessary parameter is measured unlike the normal value. For example; the temperature sensor of a hot water system measures that the temperature of the water is too high. Due to the restriction of methods and technologies of fault diagnosis systems, the location and the reason for a problem cannot be precisely determined within the components, so that the location of this problem can only be limited to the scope of the subsystem – the hot water system.

C. *Description of the automation system models*

Model based analysis is an analyzing method using modeling to perform the identification and the propagation of a problem impact [12]. As a prerequisite, the component model, the functional model and the requirement model should be well-defined by the manufacturer. The proposed models can be built in the form of Figure 2.

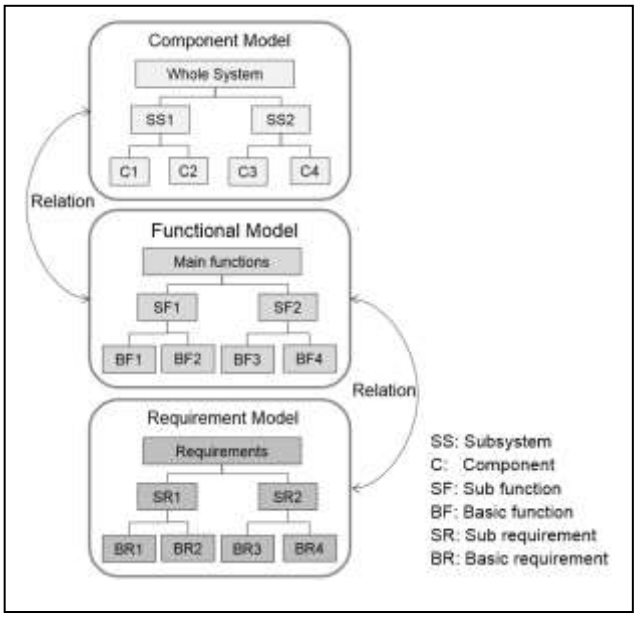

Figure 2. The graphic overview of the automation system models

The component model has the following attributes: The component ID indicates a unique identification of a component. The component name indicates the name of a component. Connected components show the other components connected with the target component. Associating subsystems describe the location of the component in the physical level. Associating functions indicate the destination of a component in this system. Activities describe the activities of a component. Therewith, the destination of it can be attained. Input describes the inputs of the component, including information and materials. Output describes the result of a component, including information and materials. Associating

Proc. of The Third Intl. Conf. on Advances in Computing, Electronics and Communication - ACEC 2015

ISBN: 978-1-63248-064-4 doi: 10.15224/ 978-1-63248-064-4-18

requirements indicate all the requirements, which must be fulfilled by the component, such as the safety dependency between the target component and the connected components.

The functional model consists of basic functions, sub functions and main functions. The following attributes are the necessary characteristics for the functions. The function ID indicates a unique identification of a function. The function name indicates the name of a function, which is the objective of a component, a subsystem or some subsystems. The containing functions describe the functions, which will be used by this target function. The associating functions indicate the functions which require this target function. The associating component indicates the components, which are needed by the target function. The associating systems indicate the subsystem which is needed by the target function. Usability indicates if the target function can be activated. It corresponds with the attribute "activating commands". Activating commands indicates the commands which activate the associating function. Associating requirement indicates all the requirements which the function muss be fulfilled in case of performing the target function. For example, the domain dependency between two functions, the next function of the function watering is the function heating. When the function watering is defect, the function heating should be not performed, because it disobey the domain dependency.

The problem propagation, which means that the effect of a problem is propagated in the automation system, is determined via the functional model at the functional level. Hence, the requirement model considers only non-functional requirements to assess the available functions. As mentioned in the literature [11], the non-functional requirements are considered as follows:

- Interface requirements, the rest functions can be operated by the user, so that the system is able to communicate with the user. That means, when the function of the interface component becomes inoperative, the functions of the other components cannot be performed.
- Security, the inoperative function and the not affected functions should not threaten the privacy data and operation. When a memory is defect that stores the system's root password, all the functions cannot be activated until the repair considering the security of the privacy of the user.
- Safety (Survivability), the rest functions must not threaten the safety of system as well as of the user.

Each element of the requirement model has the following features. The requirement ID helps to identify the target requirement. The requirement name is used for identification of the target requirement. The associating function indicates the function that should be verified by the target requirement. The containing Requirements is the constituents of the target requirement. But those requirements should be logically one level lower than the target requirement. The associating requirements need the target requirement. Safety to only singly carry out this function indicates that it would not affect the safety, when the function is performed separately.

III. Concept of identifying the available functions

The method proposes three analysis modes to identify a) the impact on the component view, b) propagate the impact on the function view, and c) verify the availability of not affected functions on the requirement view. As mentioned above, following introduction to the problem propagation analysis is based on a defective component or a defective subsystem.

The common process of problem propagation analysis is structured on three layers: component analysis, function analysis and requirement analysis (see Figure 3).

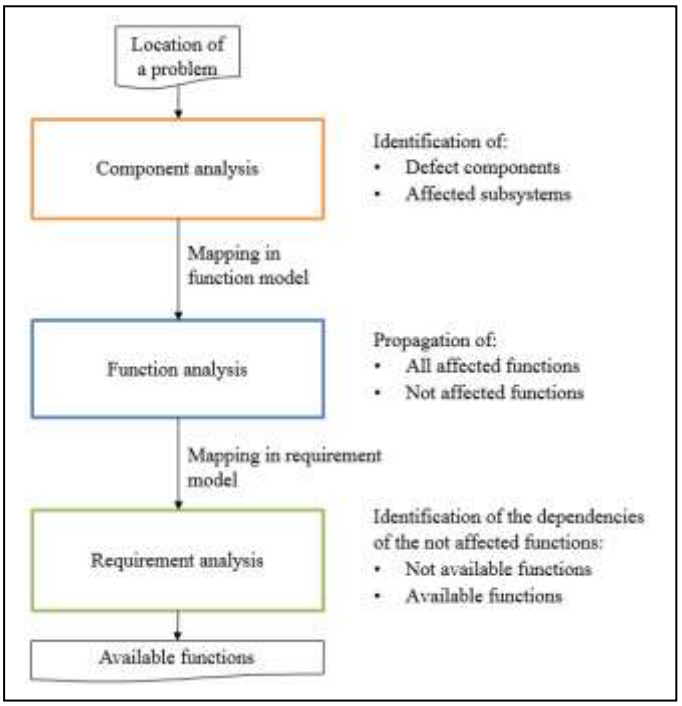

Figure 3. The procedure of the problem impact propagation analysis

Firstly, the target of the component analysis is to find out the propagation in the component model, including affected components and subsystems. Afterwards, the analysis result is mapped into the functional model by means of the relations between components and functions.

Secondly, the function analysis proposes to ascertain the propagation in the functional model, including affected basic functions, sub functions and affected main functions. Accordingly, the not affected functions are derived from the functional model. Afterwards, the analysis result is mapped to the requirement model in accordance with the relation between functions and requirements.

Finally, the purpose of the requirement analysis is to verify whether the affected functions and the not affected functions satisfy the pre-defined requirements by the manufacturer. Afterwards, the available functions for a certain problem are confirmed.

In the following section the details of the problem propagation analysis approach are discussed. Considering the diversity of fault diagnosis results, the location of a problem is divided into three cases, a defect component, two defect components, and a defect subsystem.

Proc. of The Third Intl. Conf. on Advances in Computing, Electronics and Communication - ACEC 2015

ISBN: 978-1-63248-064-4 doi: 10.15224/ 978-1-63248-064-4-18

A. *Identifying the states of components and functions*

The states of components and functions need to be identified as "defect", "normal" and "affected". Normal means that the component can perform its activities as usual. Defect means that the component cannot perform its activities. Affected means that the component is able to perform the activities, but it should be deactivated due to the special requirements, such as safety dependencies with a defective component.

For a function, there is additionally another state, "partially normal", which means that parts of child functions of the farther function are available.

B. *Impact analysis in case of defect components*

Firstly, in the scope of the component model, the affected components and subsystems are determined. By mapping the defect components to the component model. Afterward, the connected components can be confirmed, such as material connection, information connection and energy connection. According to these dependencies, the connected components are verified whether they are affected. For example, a valve is connected with an air pump. When the air pump is broken, then it will be tested if the safety of valve keeping the pressure in the milk system is dangerous. According to the safety dependency of two components, the ability of the valve depends on the functionality of the air pump, then the milk system is not safe anymore. Afterwards, the components connected to the valve in this system and other systems should be further evaluated until no more connected components are affected.

The associated subsystems with the defected and affected components in the component model are identified if the subsystem is also defect. When the defect component is an optional component for the subsystem, then the subsystem is not defect. Instead, the defect component is necessary for the subsystem, then the subsystem is defect. Consequently, using the relation between the components and the basic functions, the affected basic functions can be identified. At the same time, using the relationship between subsystems and sub functions, the possible affected sub functions can be determined.

Secondly, all the functions are described in the functional model, the possible affected sub functions are appraised, whether they are truly affected, depending on the following relationship between functions [13]:

- *Mandatory* means that the junior function is required by the superior function. When the junior function is defect, the superior function is also affected.
- *Optional* means that the junior function is optional for the superior function. Hence, when the junior function is defect, the superior function is not affected.
- *Or* means that only one of the junior function is a possible selection, and at least one of the junior functions of the same superior function should be selected. When all the junior functions for the same superior functions are defect, then the superior function is defect.
- *Alternative* means that at least one of the junior functions must be selected. When all junior functions of the same superior function are defect, the superior function is defect. But in this case, the superior function is partially available.

Affected sub functions and affected main functions are determined in the functional model. Meanwhile, the not affected functions are also confirmed. Afterwards, with determined affected and not affected functions, it needs to assess the availability of these functions. Then, these functions are mapped to the requirement model.

Thirdly, not affected functions must be tested with special requirements in the requirement model. From the bottom to the top of the requirement model, the not affected basic functions will be tested. If it wants to be singly activated, it must fulfill the corresponding special requirements, such as safety, security, standards, and special demands by the user. Require [8] as a relationship characteristic means that the function *A* in a product implies the function *B*. When the function *B* is defect, then the function *A* should not be performed. Here the attribute *require* is used for verifying the dependency among functions at the same level.

Finally, the available functions are derived from the not affected functions.

C. *Impact analysis in case of defect subsystems*

Firstly, the location of the defect subsystem is mapped in the component model. Differing from the case of the defect component, all contained components in the defect subsystem are possible to be broken, so it considers that all these components are affected. The states of the components and the subsystem are set to defect. Afterwards, the defect subsystem will be mapped to the requirement model. With the help of the variant dependencies on subsystems, the affected subsystems are confirmed. Then, the process of analysis of the boundary component, which is located in the boundary of the subsystem and connected to the components in the other subsystems, is performed. According to the dependencies on the boundary components, which connect to the components in the other subsystems. If the dependency is true, then the connected component in the other subsystems is also affected.

As a result, all the affected components, the defect subsystems and all possible affected subsystems can be determined with the help of the component model. Consequently, this result will be mapped into the functional model. Here, all sub functions associating with the defect subsystems are set as defect. Afterwards, the rest analysis process is the same with a component being defect.

IV. Evaluation of the impact analysis for an industrial coffee dispenser

Proc. of The Third Intl. Conf. on Advances in Computing, Electronics and Communication - ACEC 2015

ISBN: 978-1-63248-064-4 doi: 10.15224/ 978-1-63248-064-4-18

This section introduces the analysis process of an industrial coffee dispenser. Figure 4 shows a simplified industrial coffee dispenser. In this figure, the milk system is the focus of the system. There are still a water system and a broth system.

The hot water system has a steam boiler, which stores the hot water and the steam. The milk system consists of four components, a steam valve, an airm pump, a clamp valve and an espresso nozzle. This system can provide the milk and the milk foam. The broth system consists of a brewer and an espresso nozzle. On the one hand, the brewer is able to mix the coffee and the hot water. And on the other hand, the espresso nozzle is incharge of mixing the espresso and the milk. Here the espresso nozzle is a shared component for two systems. The hot water system has an important part, it contains the steam boiler, which can provide hot water, cold water and steam.

Figure 5 shows an example to carry out the problem impact analysis process of the industrial coffee dispenser in case of a defect air pump.

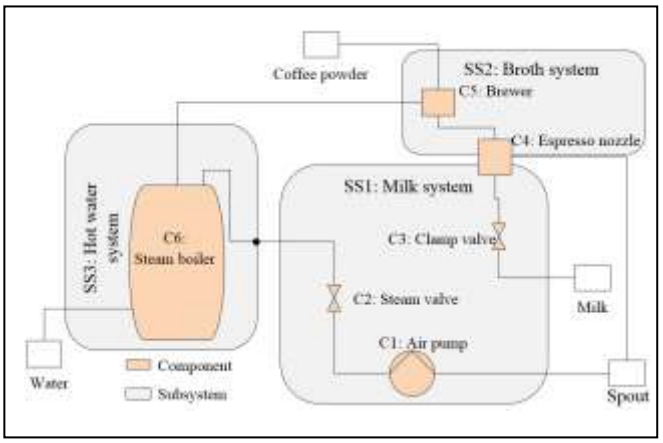

Figure 4. An scheme of a simplifying industrial coffee dispenser

In the component model, the state of all components and subsystems is set to normal. On the basis of the location of a problem is component 1 *air pump*, which is set as defect. In the component model the air pump belongs to the subsystem 1 *milk system*. Then the subsystem 1 is set as affected. According to the relationship between components and basic functions and the relation between subsystems and sub functions, the following rules are used for the reasoning:

- When Component_State is defect and the Component & Basic_Function is true, the Basic_Function is defect.
- When SubSystem_State is affected and the SubSystem & Sub_Function is true, the Sub_Function is affected.

As a result, the basic function 1 *left the luft through* is defect, meanwhile, the sub function 1 *produce milk foam* and the sub function 2 *produce milk* are possible affected.

Then, functions of different levels are analyzed in the functional model according to the result of the mapping. For one thing, the analysis process is limited to the possible affected sub functions. Afterwards, all sub functions are assecessed with the help of the affected basic functions. On the basis of the relations of sub functions and main functions, the affected main functions are inferred. The following inference rules are necessary:

- When Basic_Fun is defect and Basic_Fun & Sub_Fun is true and Defect_Basic_Fun for Sub_Fun is mandatory, then the Sub_Fun is defect.
- When Sub_Fun is defect and Sub_Fun & Main_Fun is true and Defect_Sub_Fun for Main_Fun is mandatory, then the Sub_Fun is defect.

In the industrial coffee dispenser, the sub function 1 *produce milk foam* and the main function 1 *produce capuccino* are defect. And the basic function 2 *let the steam through*, the basic function 3 *let the milk through*, the basic function 4 *let the liquid through*, the basic function 5 *mix* and the basic function 6 *store stream and water* are not affeced. The sub function 2 *produce milk*, the sub function 3 *produce espresso*, the sub function 4 *produce hot water*, the sub function 5 *produce steam* and the main function 2 *produce milk coffee* are not affected.

Finally, these not affected functions are verified by the corresponding requirements.

- When FunX is not affected and FunX_Safe_Dependency is false, then FunX is defect.

As a result , the basic function 2 *let the steam through* cannot be individuelly performed because of the safety dependency. On the contrary, the other not affected functions can be normally performed as available functions.

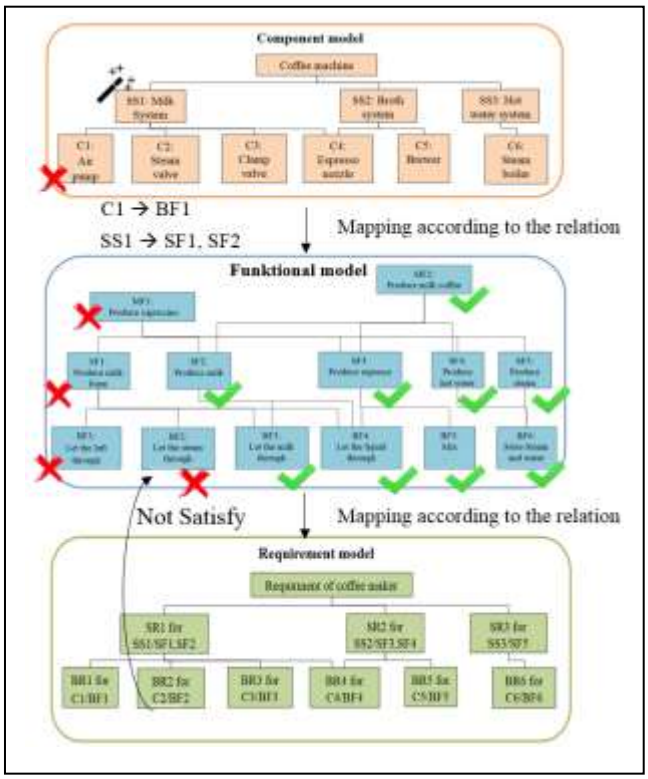

Figure 5. The procedure of problem effecet propagation analysis for an industrial coffee dispenser

v. Conclusion and future work

The presented model-based fault effect analysis approach provides an opportunity for automation systems to

ISBN: 978-1-63248-064-4 doi: 10.15224/ 978-1-63248-064-4-18

obtain the available functions in the operation stage. The proposed approach used the results of a preliminary fault diagnosis performed on the automation system, i.e. fault location of components and employs the existing requirement-function-component model, supplied by the manufacturer. The details of basic attributes of the element within the models were specified in chapter 2.

Using the fault location, all affected components are confirmed according to the component model. Afterwards, by mapping all defected components onto the functional model, all affected function are confirmed. Finally, not affected functions are verified by the non-functional requirements in the requirement model, e.g. material dependency, information dependency, energy dependency and domain dependency.

For the future work, the sequential procedure of the fault effect analysis presented above will be considered as interactional, i.e. component analysis, function analysis and requirement analysis are dependent processes. Hence, an approach is needed to efficiently obtain the available function for a fault, e.g. the application of agents or other encapsulation technologies. Moreover, the utilization of the available functions will be improved, e.g. by checking the available tasks in the real-time task list or the available solutions regarding the available functions

Acknowledgment

We thank Chinese CSC (China Scholarship Council fellowship Grant) for the financial support.

Reference

[1] P. Choe, D. T. Jeffrey, and S. Tong, "Impact of cognitive automation in a material handling system on manufacturing flexibility," International Journal of Production Economics, 2015.

[2] H. Wang, N. Jazdi, and P. Goehner, "An Agent-Based Concept for Problem Management Systems to Enhance Reliability," in Proc. 2014 Theoretical and Applied Aspects of Cybernetics, Kyiv, 2014, pp. 283-293.

[3] B. A. Clegg, A. Z. Vieane, C. D. Wickens, R. S. Gutzwiller, and A. L. Sebok, "The impacts of automation-induced complacency on fault diagnosis and management performance in process control," in: Proceedings of the Human Factors and Ergonomics Society Annual Meeting. SAGE Publications, 2014. pp. 844-848.

[4] R. Ferdous, F. Khan, R. Sadiq, P. Amyotte, and B. Veitch, "Fault and event tree analyses for process systems risk analysis: uncertainty handling formulations," Risk Analysis, Jg. 31, Nr. 1, 2011, pp. 86-107.

[5] Y. Wang, G. Cheng, H. Hu, and W. Wu, "Development of a risk-based maintenance strategy using FMEA for a continuous catalytic reforming plant," Journal of Loss Prevention in the Process Industries, Jr. 25, Nr. 6, 2012, pp. 958-965.

[6] D. M. Shalev, and J. Tiran, "Condition-based fault tree analysis (CBFTA): A new method for improved fault tree analysis (FTA), reliability and safety calculations," Reliability Engineering & System Safety, Jr. 92, Nr. 9, 2007, pp. 1231-1241.

[7] T. Kurtoglu, and I. Y. Tumer, "A graph-based fault identification and propagation framework for functional design of complex systems," Journal of Mechanical Design, Jr. 130, Nr. 5, 2008.

[8] U. Kannengiesser, and H. Muller, „Towards Agent-Based Smart Factories: A Subject-Oriented Modeling Approach," In Proceeding of Web Intelligence (WI) and Intelligent Agent Technologies (IAT), 2013 IEEE/WIC/ACM International Joint Conferences on, IEEE, Vol. 3, 2013, pp. 83-86.

[9] S. Simani, C. Fantuzzi, and R. J. Patton, Model-based fault diagnosis in dynamic systems using identification techniques; London: Springer -Verlag, 2013.

[10] M. Bordasch and P. Gohner, "Fault Prevention in Industrial Automation Systems by means of a functional model and a hybrid abnormity identification concept," in Prceeding of Industrial Electronics Society, IECON 2013-39th Annual Conference of the IEEE, IEEE, 2013, pp. 2845-2850.

[11] Ji, H., Lenord O. , and D. Schramm, A Model Driven Approach for Requirements Engineering of Industrial Automation Systems, in Proceedings of the 4th International Workshop on Equation-Based Object-Oriented Modeling Languages and Tools, Zurich, Switzerland, 2011.

[12] M. R. Maurya, R. Rengaswamy, and V. Venkatasubramanian, "Application of signed digraphs-based analysis for fault diagnosis of chemical process flowsheets," Engineering Applications of Artificial Intelligence, Jg. 17, Nr.5, pp. 501-518.

[13] P. Trinidad, D. Benavides, A. Duran, A. Ruiz-Cortes, and M. Toro, "Automated error analysis for the agilization of feature modeling," Journal of Systems and Software, vol. 81, Nr. 6, pp. 883–896, 2008.

About Authors:

Huiqiang Wang – The 3nd year PhD student, Institute of Industrial Automation and Software Engineering (IAS), University of Stuttgart, Pfaffenwaldring 47, 70569, Stuttgart, Germany; Major Fields of Scientific Research: reliability and intelligence, availability, education on electronics.

Nasser Jazdi – Scientific staff member, Institute of Industrial Automation and Software Engineering (IAS), University of Stuttgart, Pfaffenwaldring 47, 70569, Stuttgart, Germany; stuttgart.de Major Fields of Scientific Research: Software Reliability, learning aptitude for industrial automation, Soft Computing for Industrial Automation

Michael Weyrich – Director of the Institute of Industrial Automation and Software Engineering (IAS), University of Stuttgart, Pfaffenwaldring 47, 70569, Stuttgart, Germany; -stuttgart.de Major Fields of Scientific Research: agent-oriented concepts for the industrial automation, user-oriented automation, energy optimization of technical systems, learning ability and reliability of automated systems, and concepts for the reuse in the industrial automation.