# Context-Aware System Analysis: Introduction of a Process Model for Industrial Applications

Patrick Rosenberger[1], Detlef Gerhard[1] and Philipp Rosenberger[2]
[1]Institute for Engineering Design and Logistics Engineering, TU Wien, Getreidemarkt 9/307, 1060, Vienna, Austria
[2]Department of Technical Management, University of Applied Science FH Campus Wien, Favoritenstraße 226, 1100, Vienna, Austria

Keywords: Context-Aware System Analysis, Context-Aware System Development, Contextual Process Model, Context-Activity Matrix.

Abstract: The ability to record environmental conditions with sensors enables the development of applications that can react to changing situations. These context-aware systems provide major benefits for their users, as they allow the customization of their functionalities while decreasing the need for user interaction at the same time. Despite these advantages, context-awareness still plays a minor role in industrial applications due to reasons such as the increased efforts required during the development or the absence of tools to handle the inherent complexity. To facilitate the implementation of context-aware systems, this publication introduces a process model for analysing contextual requirements and defining contextual functionalities. The approach allows the integration of context-awareness into systems and can be used in combination with all common software development methodologies. Further, a method is proposed that reduces the complexity of context-aware systems to a manageable level. To demonstrate how the presented approaches can be applied, this paper finishes by showing the development a context-aware information system for the workers at the shop-floor of an injection moulding company.



## 1 INTRODUCTION

During the last decades, sensors and smart devices were deployed extensively in manufacturing companies (Gungor and Hancke, 2009). This factor enabled the development of systems that can recognise change and act upon (Gröger and Stach, 2014; Monostori, 2014). These so-called "context-aware" applications (Biegel and Cahill, 2004) offer new possibilities in assisting their users while decreasing the necessary user interaction at the same time. Instead of requiring the operators to choose the needed functionalities, context-aware systems predict the desired services and execute them automatically (Alegre et al., 2016). Despite their potential, the deployment of these applications is low due to the increased efforts needed for the development (Fahy and Clarke, 2004) or the diversity of data sources (Chen et al., 2014).

To support the elicitation of user requirements and encourage the deployment of context-aware systems, this publication addresses two major problems. First, traditional project development models do not meet the requirements of context-aware systems. Therefore, developers have to come up with their own "try and error" approaches. This strategy is likely to increase the overall project duration and costs and can even lead to the failure of the project. Secondly, the number of contexts and context-aware activities can be extensive even for small systems. This can become a problem in later development stages if the same contexts are used to identify different activities.

The outline of this publication is as follows: Section 2 starts with the introduction of a process model for analysing and defining contextual functionalities. The model is used in addition to conventional techniques, so that the developers can rely on traditional approaches to design the base system and use the proposed model to incorporate contextual features. To deal with the high number of contexts used by these systems, section 3 proposes the context-activity matrix, a lightweight tool for handling the complexity that is inherent in most context-aware systems. To show the applicability and practicality of the proposed solutions, section 4



Rosenberger, P., Gerhard, D. and Rosenberger, P.
Context-Aware System Analysis: Introduction of a Process Model for Industrial Applications.
DOI: 10.5220/0006802703680375
In *Proceedings of the 20th International Conference on Enterprise Information Systems (ICEIS 2018)*, pages 368-375
ISBN: 978-989-758-298-1

presents an industrial use case: a context-aware information system is designed for the shop-floor of an injection moulding company. Finally, section 5 finishes with a critical discussion of the results.

## 2 CONTEXT-AWARE SYSTEM ANALYSIS

The analysis of context-aware systems requires additional efforts that are not covered by traditional process models. To encounter this problem, many research activities on issues such as customized eliciting techniques (Sitou and Spanfelner, 2007; Choi and Lee, 2012) or adapted contextual models (Choi, 2008; Ruiz-López et al., 2013) have been carried out. Despite their high relevance, these publications were mainly focused on specific problems and not the broader process. Some publications that addressed this topic (Hong et al., 2005; Choi, 2007; Desmet et al., 2007; Alegre, 2016) only considered it as a side issue. Therefore, a general process model that characterizes the analysis of context-aware system requirements is still missing.

To fill this gap, this section introduces a holistic method to elicit and define context-related system requirements. The proposed process model should not be seen as a stand-alone approach, but should rather be used as an extension to already existing system development methods. Using the model, the following key questions are addressed:

- Which activities can be enhanced by using context?
- How can these activities be enhanced?
- What consequences does this have on the business process?
- Which contexts can be used to identify the activities?

The proposed process model divides the analysis and design activities of context-aware systems in three phases:

- **Activity determination:** identification of activities that can be assisted by contextual functionalities.
- **Process definition:** definition of how the system should react to the occurrence of a context.
- **Context elicitation:** determination of the contexts.

Figure 1 shows the three phases including their activities. While the process model separates the process definition and the context elicitation phases, their tasks can also be carried out concurrently (see section 2.4).

### 2.1 Activity Determination

The first step in developing a context-aware system is to identify the activities that can be supported contextually. Therefore, all tasks covered by the system must be examined to decide whether contextual features could assist the users. A complete system description highlighting all contextual-assisted activities should be created. To improve comprehensibility, activity diagrams emphasising contextual-enhanced tasks could be created using modelling languages like UML or SysML.

### 2.2 Process Definition

Based on the results of the activity determination, the system's behaviour and responses must be defined. The goal is to further develop the system and achieve an agreement between all stakeholders. As stated by Perera et al. (2014), the different

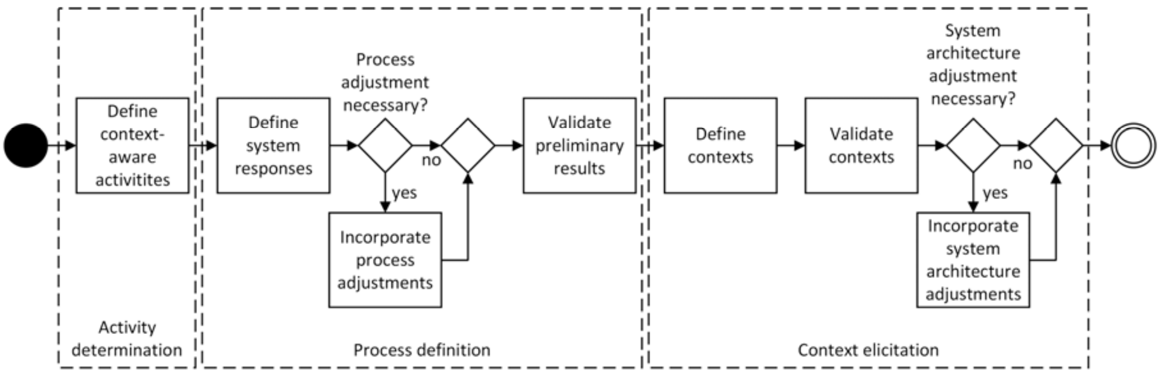


Figure 1: Contextual elicitation and analysis: process model.

actions can be dived into two categories:

- **Active execution:** a predefined functionality gets executed automatically.
- **Passive execution:** a set of options is proposed from which the user can choose the preferred one.

In case a system response triggers a new process flow, changes of the systems functionalities must be incorporated. This also requires the definition of how the changes affect the users' and stakeholders' working procedures. Regardless of the need for a system adaption, the process definition phase ends with the validation of the preliminary results.

## 2.3 Context Elicitation

Next, the contexts that identify the activities need to be defined. Forming a central part of the contextual requirements elicitation, many different approaches have been proposed (Hong et al., 2005, Kolos-Mazuryk et al., 2005, Munoz et al., 2006). In general, the approaches can be divided into three categories (Alegre, 2016):

- **Category-based elicitation:** the developers list all measurable context types and then determine the present ones for each activity.
- **Model-centred elicitation:** the developers select the contexts by modelling a context-unaware version of the system and then define which contexts are needed to provide the intended functionality.
- **User-centred elicitation:** the developers determine the contexts by interviewing the users or other stakeholders.

The approach suited best for industrial applications strongly depends on the use case. In case of highly deterministic process flows, the existing documentation contains all needed information to define the contexts. If we consider the automotive industry where each manipulation is planned carefully, the contextual enhancement can be derived directly from existing process descriptions using the category-based or model-centred elicitation approach. If, on the other side, the process flows are not deterministic, the existing documentation will not be sufficient to fully define the different contexts. Therefore, user-centred techniques are suited best.

After defining the contexts, they must be validated to ensure the following properties:

- **Uniqueness:** The contexts describing each activity must be unique. Otherwise, the system will not be able to differentiate between activities.
- **Measurability:** All contextual data points used by a context must be measurable. If not, the system will not be able to sense related contexts.

If one of these conditions is violated, the system will not be able to operate in the intended way. Therefore, the adaption of the system architecture might be required. By introducing new elements such as sensors, additional contextual data can be acquired ensuring the uniqueness and measurability of all contexts. If for example the state of a machine should be measured without the availability of internal sensors, external sensors can be used.

## 2.4 Joined-phase Elicitation for the User-centred Approach

As stated above, it is possible to combine the tasks of the process definition and context elicitation phases (see Figure 2). By doing so, the determination of the system's contexts and responses

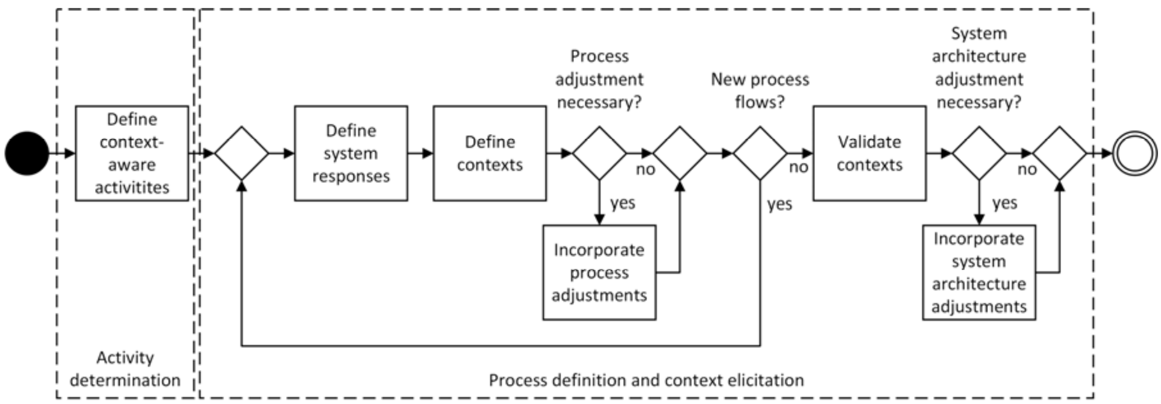


Figure 2: Contextual elicitation and analysis: joined-phase elicitation.

Table 1: Context-activity matrix: basic structure.

| *Identifier* | **Context type 1** | **Context type 2** | **Context type 3** | **…** |
|---|---|---|---|---|
| **Activity 1** | Contextual data 1/1 | Contextual data 1/2 | Contextual data 1/3 | … |
| **Activity 2** | Contextual data 2/1 | Contextual data 2/2 | Contextual data 2/3 | … |
| **Activity 3** | Contextual data 3/1 | Contextual data 3/2 | Contextual data 3/3 | … |
| **…** | … | … | … | … |

are performed simultaneously, reducing the time needed for these tasks. This is useful in case of a user-centred elicitation, as the time and efforts required from the users are decreased. Despite its advantages, the condensed model is more error-prone due to the missing intermediate validation. Therefore, we discourage unexperienced developers to use this approach.

Following the activity determination, the system's response and the corresponding contexts are determined concurrently. If necessary, process adjustments are performed to align the system to the changed requirements. As the adjustments are made after the context elicitation, it is necessary for each newly introduced process flow to repeat the response and context definition steps. Finally, the validations are combined and if necessary the system architecture is adjusted which leads to the process model shown in Figure 2.

# 3 CONTEXT-ACTIVITY MATRIX

Validating middle- and large-sized systems can be challenging due to the high number of contexts that need to be considered. To facilitate this task, we introduce the context-activity matrix (see Table 1). Using this method, the sensed contexts and the underlying contextual data can be mapped clearly, providing a comprehensive overview over the context-awareness of an application. The matrix can be considered as extension to the work of Omasreiter and Metzker (2004) who used a slimmed approach to identify dangerous driving situations that require an automatic adaption of the cruise control. While their work focused on identifying critical circumstances, we enhance the matrix by integrating different levels of abstraction and by easing the validation of uniqueness and measurability.

## 3.1 Basic Structure

The construction of the matrix starts by labelling the rows with the contextually assisted activities and the columns with the sensed context types. The specific contextual data that makes up the individual contexts is noted in the remaining fields. Finally, the top left field is used to state the identifier that allows the differentiation between different matrices. This approach is quite different to the work of Omasreiter and Metzker (2004) and we encourage the usage of this layout for two reasons. First, in most cases a limited set of context types is used to identify the activities. This allows the usage of the same header for different matrices. Secondly, this layout enables the comparison of the contextual data. Their uniqueness can be validated by comparing the individual rows. The measurability can be secured by ensuring that the system has the needed sensors and functionalities to measure each contextual data.

## 3.2 Levels of Abstraction

As the number of activities and context types can be extensive, introducing different levels of abstraction can help to improve the clarity. While there is no restriction on splitting the matrix, we encourage to follow the use case breakdown. This provides a clear line for separation and ensures that the subdivision is aligned with other development approaches. Furthermore, context types can be used for separation. If for example different locations are mapped, each matrix can contain the activities of one location. We also encourage using separate matrices for specialized applications with a unique set of contexts. For example, monitoring the characteristics of a machine with internal sensors results in a specialized contextual data set. Therefore, a clean separation from the other matrices is possible.

# 4 USE CASE

In the following, the usage of the process model and the context-activity matrix are demonstrated in an industrial use case. Within the research project FACTS4WORKERS (Facts4Workers, 2017), a context-aware information system is developed for the industrial partner Thermolympic. The objective is to assist the workers of the production area in their daily tasks. The company's shop-floor workforce can be separated into two groups: operators and team

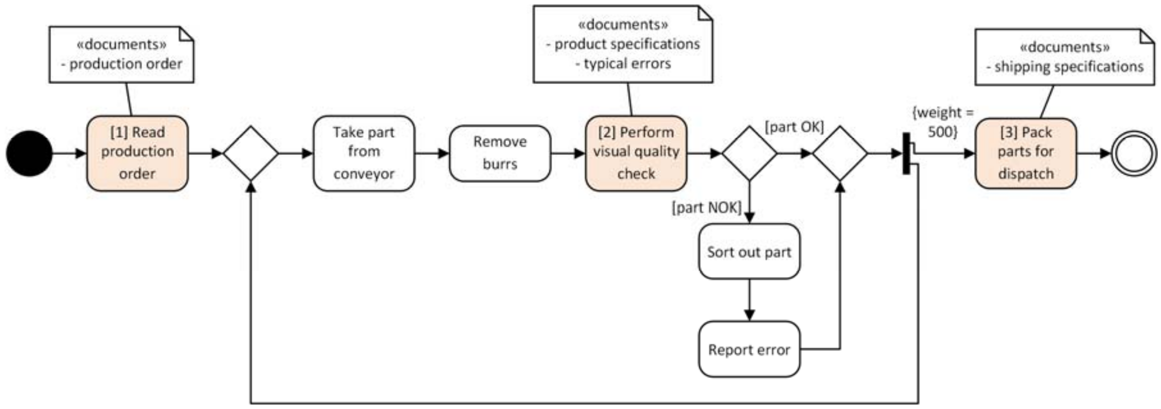


Figure 3: Process flow: operator.

Table 2: Context-activity matrix: operator.

| *Operator contexts* | Role | Location | Activity | Time |
|---|---|---|---|---|
| **[1] Display production order** | Operator | Machine location including up to 1m distance | | At the start of the shift |
| **[2] Display product specifications and typical errors** | Operator | Machine location including up to 1m distance | | |
| **[3] Display dispatch specifications** | Operator | Packaging area | Previously processed product | |

leaders. Operators are responsible for deburring the produced parts and performing visual quality controls. If a part does not pass the test, it is sorted out and the error is recorded. Additionally, they prepare the parts for dispatch by packing them according to the customer preferences. During their daily tasks, the operators must be supplied with different work instructions like the production order, the product specifications, a listing of typical errors and the dispatch specifications. Team leaders act as supervisors who monitor the operators' tasks and are the contact person in case of problems or uncertainties. Additionally, they are responsible for setting up the machines at production start, change the machine parameters if environmental conditions like temperature or humidity are changing and repair the machines in case of errors. Therefore, they must be provided with the production orders, machine documentation as well as error and change reports. To assist the workers, a context-aware information provision system that can sense the different roles as well as the users' location and activities is needed. Due to the extent of the use case, the analysis will be reduced to the central elements of the system.

## 4.1 Operator

During their work, the operators must perform visual controls, sort out faulty parts, record errors and prepare good parts for dispatch. A detailed model stating the different activities can be seen in Figure 3. To ease understanding, the contextual assisted activities are highlighted.

Following the process model stated in section 2, the analysis starts by identifying the activities the context-aware system is going to assist. For each of the selected, tasks it is specified which functionalities are needed to provide benefits to the users. Building the base of the context-activity matrix, the system responses are recorded in the first column of the context-activity matrix. By assigning each line with an identifier, the tractability between the matrix and the process flow diagram is ensured. As no new processes are triggered, an adaption of the process flow is not necessary. After validating the intermediate state with the workers and management, the contexts are identified. Due to the high degree of flexibility the operators have, we choose a user-centred elicitation approach.

First, we describe the intention of the context-aware system and how it would work. Then we assist the workers in defining the contexts and contextual data. After listing the results in the context-activity matrix, a final validation is performed.

As shown in Table 2, the four context types role, location, activity and time are used. Playing a key part, the role is used to differentiate the activities of

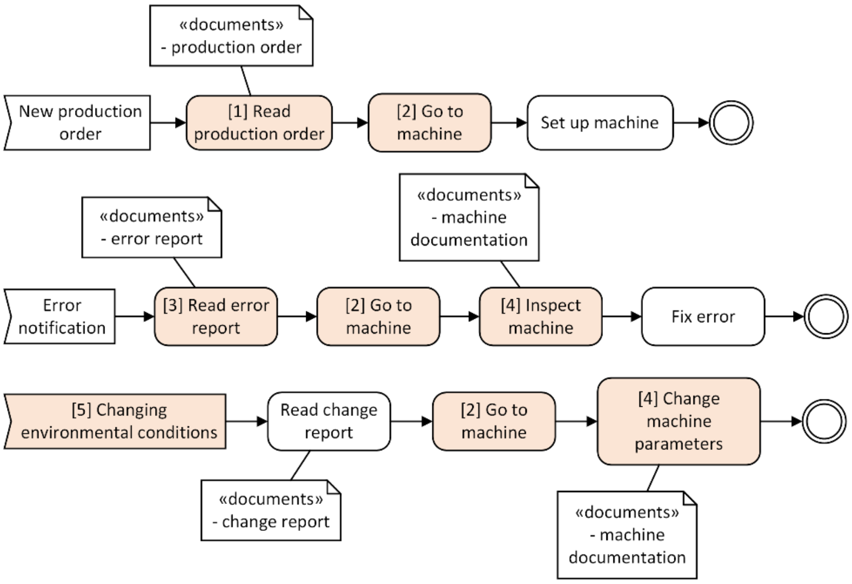


Figure 4: Process flow: team leader.

Table 3: Context-activity matrix: team leader.

| ***Team leader contexts*** | **Role** | **Location** | **Activity** | **Environmental conditions** |
|---|---|---|---|---|
| **[1] Display production order** | Team leader | | New production order | |
| **[2] Provide navigation** | Team leader | Current location + movement | | |
| **[3] Display error report** | Team leader | | New error | |
| **[4] Display machine documentation** | Team leader | Machine location including up to 1m distance | | |
| **[5] Detect environmental condition changes** | Team leader | | | Temperature + humidity |

the operators from other users. As each machine produces a specific part, the machine location is used to distinguish between the different products. This way one operator can work on different machines and still receive the right information automatically. An exception is the display of the dispatch specifications. Here the systems recognize the changed requirements when the worker enters the packaging area. As one worker can process different parts, the correct information is determined by retrieving the last handled product. While reporting an error could also potentially be supported contextually, another system is used to do so. Therefore, this activity is not considered.

## 4.2 Team Leader

Team leaders are assisted in daily tasks like setting up the machines, changing their parameters, and repairing them in case of a malfunction. The different process flows can be seen in Figure 4. Again, the contextual assisted activities are highlighted.

The analysis starts with the identification of the contextual assisted activities and the definition of the system reactions. As changing environmental conditions are triggering a new process flow, the process steps are modelled, and the analysis is repeated. Then, the context-activity matrix is created. To decrease the matrix size and increase clarity, we choose to split it up. The context type "role" is used to differentiate between the two user groups, assuring unique contexts. Finally, the contexts are defined, and the results are validated by using the user-centred elicitation approach.

Table 3 shows the corresponding context-activity matrix. Whenever the production of a new product is initiated, the production order is forwarded to the team leader automatically. He/she then inspects the document and goes to the machine. If not already requested, the system detects his/her movements and provides navigation. In case of an error, the system forwards the error report. Again, navigation is provided upon request or movement. As soon as he/she arrives to the machine, the corresponding machine information is displayed. In case of changing temperatures or humidity exceeding a predefined threshold, the system notifies the team leader. As producing plastic parts is dependent on these conditions, this allows him/her to react before faulty parts occur. Again, the team leader is assisted by navigation and the display of the machine documentation.

## 5 CRITICAL DISCUSSION AND CONCLUSION

As context-aware systems extend the functionalities of classical software systems, their development requires higher efforts. Nevertheless, the benefits, such as the increased usability or a decreased need for user interaction, easily surpass the downsides. This applies particularly to industrial applications since the user interaction is not always easy. For example, work clothes can prevent the usage of touchscreens, or dirt and noise can complicate the information intake. Therefore, automatically providing the needed information and functionality can result in an increased efficiency and worker satisfaction.

To encounter the disadvantage of requiring more complex development activities, the previously proposed methods offer a convenient way to facilitate context-aware system development projects. As shown in the use case implementation, the proposed process model provides a standardized way for defining context-related activities, determining system reactions and identifying individual contexts. Further, the usage of the context-activity matrix helps structuring different contexts and assures their uniqueness and measurability throughout the use case. This ensures that identified contexts can be sensed and do not overlap.

While the application of the proposed approaches has shown promising results, their benefits must be examined further. To make a general statement about their efficiency, a variety of real-world tests are required.

The application of the methods on a broader scale will also allow to detect domain specific issues that require an adaption or extension. Especially for the context-activity matrix we expect varying requirements in different areas.

To improve the methods, we encourage developers to test and adapt the approaches to their needs. Having an active community using these methods will lead to their fast evolvement and the extensive deployment of context-aware systems at industrial companies.

## ACKNOWLEDGEMENTS

The project on which this publication is based has received funding from the European Union's Horizon 2020 research and innovation program under Grant Agreement n˚63677.

## REFERENCES

Alegre, U., Augusto, J. C. and Clark, T. (2016), "Engineering context-aware systems and applications: A survey", *Journal of Systems and Software*, Vol. 117, pp. 55-83.

Biegel, G. and Cahill, V. (2004), "A framework for developing mobile, context-aware applications", *Proceedings of the Second IEEE Annual Conference on Pervasive Computing and Communications*, pp. 361-365.

Chen, M., Mao, S. and Liu, Y. (2014), "Big data: A survey", *Mobile Networks and Applications*, Vol. 19, No. 2, pp. 171-209.

Choi, J. (2007), "Context-driven requirements analysis", *Computational Science and Its Applications-ICCSA 2007*, pp. 739-748.

Choi, J., (2008), "Context: From Birth to Design", *2008 International Conference on Advanced Language Processing and Web Information Technology*, pp. 347-352.

Choi, J., and Lee, Y. (2012), "Use-case driven requirements analysis for context-aware systems", *Computer Applications for Bio-technology, Multimedia, and Ubiquitous City*, Springer, Berlin, Heidelberg, pp. 202-209.

Desmet, B., Vallejos, J., Costanza, P., De Meuter, W. and D'Hondt, T. (2007), "Context-oriented domain analysis", *Proceedings of the Sixth International and Interdisciplinary Conference on Modeling and Using Context CONTEXT 2007*, pp. 178-191.

Facts4Workers (2017), *The Goals of FACTS4WORKERS*, Facts4Workers. Available at: http://facts4workers.eu/index.php/the-goals/ (accessed at 27.11.2017).

Fahy, P. and Clarke, S. (2004), "CASS-a middleware for mobile context-aware applications", *Proceedings of the Workshop on Context Awareness within MobiSys'04*, pp. 1-6.

Gröger, C. and Stach, C. (2014), "The mobile manufacturing dashboard", *Proceedings of the IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*, pp. 138-140.

Gungor, V. C. and Hancke, G. P. (2009), "Industrial wireless sensor networks: Challenges, design principles, and technical approaches", *IEEE Transactions on industrial electronics*, Vol. 56 No. 10, pp. 4258-4265.

Hong, D., Chiu, D. K. and Shen, V. Y. (2005), "Requirements elicitation for the design of context-aware applications in a ubiquitous environment", *Proceedings of the 7th international conference on Electronic commerce - ICEC 2005*, pp. 590-596.

Kolos-Mazuryk, L., Poulisse, G. J. and van Eck, P. A. T. (2005), "Requirements engineering for pervasive services", *Second Workshop on Building Software for Pervasive Computing*, San Diego, California, USA, 2005, pp. 18–22.

Munoz, J., Valderas, P., Pelecha, V. and Pastor, O. (2006), "Requirements engineering for pervasive systems. a transformational approach", *Proceedings of the 14th IEEE International Requirements Engineering Conference*, pp. 351-352.

Monostori, L. (2014), "Cyber-physical production systems: Roots, expectations and R&D challenges", *Procedia CIRP*, Vol. 17, pp. 9-13.

Omasreiter, H. and Metzker, E. (2004), "A context-driven use case creation process for specifying automotive driver assistance systems", *Proceedings of the 12th IEEE International Requirements Engineering Conference*, pp. 334-339.

Perera, C., Zaslavsky, A., Christen, P. and Georgakopoulos, D. (2014), "Context aware computing for the internet of things: A survey", *IEEE Communications Surveys & Tutorials*, Vol. 16 No. 1, pp. 414-454.

Ruiz-López, T., Rodríguez-Domínguez, C., Rodríguez, M. J., Ochoa, S. F., and Garrido, J. L. (2013), "Context-aware self-adaptations: From requirements specification to code generation", *International Conference on Ubiquitous Computing and Ambient Intelligence*, pp. 46-53.

Sitou, W., and Spanfelner, B. (2007), "Towards requirements engineering for context adaptive systems", *31st Annual International Computer Software and Applications Conference (COMPSAC 2007)*, pp. 593-600.