Available online at www.sciencedirect.com

ScienceDirect

Procedia CIRP 91 (2020) 427–432

www.elsevier.com/locate/procedia

30th CIRP Design 2020 (CIRP Design 2020)

The CASAD Matrix Method: Introduction of a Technique for the Documentation, Analysis, and Optimization of Context-Aware Systems

Patrick Rosenberger[a,]*, Manfred Grafinger[a], Detlef Gerhard[b], Martin Hennig[a], Stefan Dumss[a]

[a] *TU Wien, Institute of Engineering Design and Product Development, Getreidemarkt 9/307, 1060 Vienna, Austria*
[b] *Ruhr University Bochum, Digital Engineering Chair, Universitaetsstr. 150, 44801 Bochum, Germany*

* Corresponding author. Tel.: +43-1-58801-30767; fax: +43-1-58801-30798. *E-mail address:* patrick.rosenberger@tuwien.ac.at

Abstract

The availability of precise sensors and the extensive usage of smart devices enable the development of systems that can sense their environment and recognize specific contexts. In recent years, so-called context-aware systems gained increasing popularity as they can provide situational assistance and automate nondeterministic processes. Despite their advantages, these systems still play a minor role in industrial applications due to reasons such as the increased efforts required for the engineering and development process and the absence of methods to handle the inherent complexity. This paper introduces the CASAD matrix method that facilitates the development of context-aware systems and reduces their complexity to a manageable level. The matrix is based on the design structure matrix method of Steward [1] and allows the structured documentation, analysis, and optimization of context-related systems properties. Further, different approaches for analyzing the matrix are shown and their outcomes discussed. Finally, the practical application is demonstrated in an industrial use case. Thereby, the CASAD matrix method is used to document, analyze, and optimize the contextually assisted activities, the contextual services, and the contexts of a context-aware assistance system for the shop floor of an injection molding company.

Keywords: Context-aware systems; ubiquitous computing; design structure matrix method; industry 4.0

1. Introduction

With the increasing deployment of sensors and the usage of smart devices in all aspects of life, systems became able to precisely sense their environment, recognize specific contexts, provide situational assistance, and automate nondeterministic processes [2]. Compared to the proliferation of context-aware applications in the private end-user sector (B2C), contextual information is hardly used in industrial applications (B2B) [3]. Exceptions can only be found for isolated domains like robotic vision [4] or condition monitoring in real-time [5]. Alegre et al. [6] conducted a study among 750 researchers from science and industry to determine the cause of this deviation. The authors identified the lack of methods and techniques to support the documentation, analysis, and optimization of context-related system properties as an urgent issue that impedes the development of large-scale context-aware systems. Compared to smartphone or smart home applications, industrial systems are usually characterized by a broader scope, extended functionalities, and most importantly a higher complexity. Therefore, their development requires adapted methods that reduce this complexity to a manageable level [7].

In the following, the paper introduces a method that enables the documentation, analysis, and optimization of context-related system properties. The method is based on the design structure matrix method of Steward [1] and follows the basic idea of labelling interactions between system components, organizational units, or processes. However, the proposed method focuses on context-related system properties and the interrelations between them.

The outline of this paper is as follows: Section 2 starts with a state-of-the-art analysis of existing techniques for the support

10.1016/j.procir.2020.03.106

of context-related system analysis activities. Afterwards, section 3 defines the term context and discusses related concepts. Section 4 introduces the CASAD matrix method and presents the basic structure, its behavior, and common applications. In section 5, different approaches for analyzing the outlined context-related system properties and their outcomes are discussed. Finally, section 6 presents the application of the method in an industrial use case. Thereby, a context-aware assistance system for the shop-floor workers of an injection molding company is developed.

2. State of the art

Based on a comprehensive survey, Alegre et al. [6] state that most of past research activities in the domain of context-aware systems have been focused on issues related to the technical implementation. In comparison, the system analysis and design play no or only a subordinate role, leading to a substantial research gap with respect to supporting methods. These findings are confirmed by the survey of Perera et al. [3], who emphasize that future research challenges are connected to the development of such methods. To support the analysis and design of context-aware systems, Wei and Chan [8] present an overview of the currently available sensor types and the contextual attributes they sense. Using this summery as a reference provides a starting point for the definition of a system's contexts. The work of Rosenberger and Gerhard [9] provides an overview of common types of contexts. This includes a discussion of their sources, durability, change rates, and degree of usage, which can serve as a starting point for the definition of the contexts. With respect to methods supporting the documentation, analysis, and optimization of context-related systems properties, Omasreiter und Metzker [10] propose the context-driven use case creation process. This approach models situations that require contextual assistance while driving with the help of use case diagrams. In addition, the contexts are documented using a matrix-shaped table. Despite its potential, the table is only focused on the documentation and must be used in combination with the corresponding use case diagrams. Independent of this approach, Choi and Lee [11] demonstrate how to apply use case diagrams for modelling context-aware system properties. The authors use matrix-shaped decision tables for documenting contextual services. Again, the matrix is designed for documentation purposes only. Based on these papers, Rosenberger et al. [12] propose the context-activity matrix for documenting the contextually supported activities, the contexts, and their relations. Further, the authors suggest using different levels of abstraction to model different perspectives on a context-aware system. Like before, the main objective of the matrix lies in documenting the system properties, so that no further analysis is conducted.

3. Contexts and related concepts

Defining the concept of contexts is not a trivial task due to the vagueness of the term, which led to the proposal of a variety of divergent definitions. In general, we acknowledge the one of Dey et al. [13] as most useful for characterizing contexts across different research areas, despite its shortcomings when applied to industrial applications [9]. Based on the works of Dey et al. [13] and Rosenberger and Gerhard [9], we define a context as "*the complete set of information that can be used to unambiguously identify an entity, its condition, and/or its situation, if this information is relevant for the interaction between a user and a system or between different systems. An entity can be a person, a place, an object, an application, or a system, including the user and the system(s) themselves*". Inextricably linked with this definition are the three concepts of contextual attributes, contextual values, and context classes:

- **Contextual values** correspond to the concrete data points that a context-aware system captures and processes to sense the current context. As an example, a machine's operating temperature of 87°C corresponds to the contextual value "87".
- **Contextual attributes** categorize the contextual values of the same category, structure, and origin. As an example, the attribute machine temperature includes the unit "°C" and all realistic temperatures.
- **Context classes** cluster contexts that are formed by the same contextual attributes. An example would be all contexts consisting of the machine temperature and the product quality of an injection moulding process.

Figure 1 shows the relationship between contexts, contextual values, contextual attributes, and context classes. A context describes a specific situation in a format that systems can collect, analyse, and process. Each context consists of one or more contextual values, and each contextual value characterizes one or more contexts. Contextual attributes group related contextual values. Each contextual value belongs to exactly one contextual attribute and each contextual attribute consists of at least two values (e.g. True / False). Context classes group contexts with the same contextual attributes. Thereby, a context belongs to exactly one class and a class consists of at least two contexts.

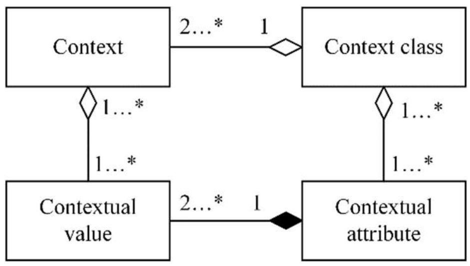

Fig. 1. Contextual attributes, contextual values, and context classes.

Greenberg [14] notes that there are three major challenges with respect to the system analysis: the definition of the contexts, the definition of the contextual services, and the definition of the contextually assisted activities. The definition of the contexts covers the elicitation of the contextual attributes and values that characterize each context. The definition of the contextual services covers the elicitation of the services that are executed or provided by a context-aware system upon the occurrence of a context. The definition of the contextually supported activities covers the elicitation of the activities in which the support by a contextual service adds value for the interaction partners.

4. The CASAD matrix method

4.1. Design structure matrix method

The CASAD matrix is based on the design structure matrix method (Fig. 2), a modelling technique that maps the elements of a system and their interactions with the help of a square matrix. A typical example would be to compare the components of a motor, whereby the interactions are used to state the parts that interact directly. Originally proposed by Steward [1], the method is mainly used to define system architectures. Compared to other network modelling techniques, the design structure matrix provides a compact and scalable representation that can be used domain-independent [15].

The creation spans three steps: First, all elements are listed along the horizontal and vertical axis. Then, their relationships are defined. Thereby, each mark states that the element on the vertical axis interacts with and depends on the element on the horizontal axis. Finally, the so-called partitions are defined. These areas of high interrelatedness represent the smallest possible unit of elements that must be considered together. To improve clarity and structuring, the matrix can be split into different sub-matrices along the partitions.

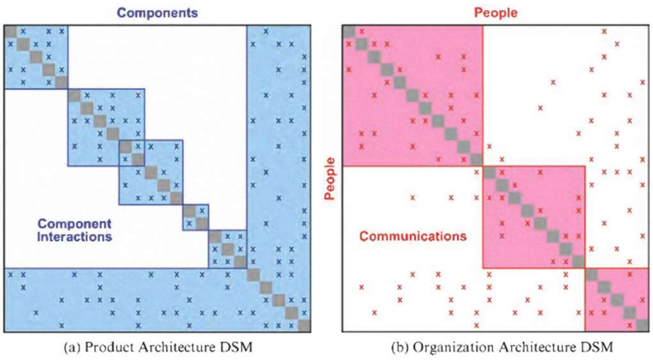

Fig. 2. Design structure matrix method [15].

Using the method during the system analysis yields the following benefits [15]:

- **Insight:** During the creation of the matrix, a deep study of the properties takes place. This leads to an increased knowledge about the system, its mode of action, its impact, and the benefits for its interaction partners.
- **Documentation:** The structured representation allows a clear and understandable representation of the context-related system properties.
- **Visibility:** The structured and compact arrangement allows a clear presentation of relevant relationships and correlations.
- **Analysis:** The mapping enables a variety of analyses that help to understand and optimize the system.
- **Scalability:** The matrix can be scaled arbitrarily, allowing a representation with different degrees of detail.
- **Extensibility:** The method can be used to map a variety of properties, going beyond the ones stated in this paper.

4.2. From the design structure to the CASAD matrix method

The CASAD matrix can be seen as extension to the design structure matrix by Steward [1] and is characterized by the same objective, creation process, and benefits. However, in contrast to the original method, the CASAD matrix method does not compare a system's or product's components but focuses on the context-related elements of context-aware systems. Further, the method does not compare the interactions between the elements among themselves, but instead maps the elements and interactions of different context-related system properties. As an example, the comparison of a system's contextual services (e.g. stop production, notify maintenance worker) and its contexts (e.g. machine failure) provides a detailed overview of the services that are executed after the occurrence of a context.

4.3. Basic structure of the CASAD matrix method

The CASAD matrix method is constructed by defining the context-related properties along the horizontal and vertical axis. It can be useful to describe the elements using keywords or IDs to avoid overloading. If several matrices are constructed, each one should be assigned with a unique ID in the first cell on the top left. The remaining cells are used to mark the relationship between the elements. Figure 1 shows the basic structure. Thereby, the content and marks are arbitrary and only intent to present the notation.

Table 1. Basic structure.

Matrix ID	Property 2 / Element 1	Property 2 / Element 2	Property 2 / Element 3	…
Property 1 / Element 1	X		X	…
Property 1 / Element 2	X	X		…
…	…	…	…	…

In addition to simple classifications, more complex mappings can be provided by substituting the "X" with a numerical or categorical value and stating a matching scale. Common examples are numbers, symbols, or colour codes. Combining different values allows the creation of high dimension matrices that comprise a variety of information. Examples for such categories are:

- The importance of activities for meeting the overall process goals.
- The importance of the contextual services.
- The sequence that the contextual services are provided.
- The quality of contexts, contextual attributes, or contextual values.

4.4. Core matrices

The **CASAD activity service matrix** as shown in table 2 is used to define the relationship between the contextually supported activities and the contextual services. This provides a detailed mapping of the services that are executed or provided during an activity as well as the activities that are supported by a contextual service. As an example, the maintenance (activity 1) and error-handling (activity 2) can both be assisted by the automatic provision of a machine's manual (service 1).

Table 2. Basic structure of the CASAD activity service matrix.

ASM ID	Service 1	…
Activity 1	X	…
Activity 2	X	…
…	…	…

The **CASAD activity context matrix** as shown in table 3 is used to define the relationship between the contextually supported activities and the contexts. This provides a detailed mapping of the contexts that characterize an activity as well as the activities that are characterized by a context. As an example, the maintenance (activity 1) can be identified by the machine ID, maintenance schedule, and date (context 1) or by the machine ID and maintenance request (context 2).

Table 3. Basic structure of the CASAD activity context matrix.

ACM ID	Context 1	Context 2	…
Activity 1	X	X	…
…	…	…	…

The **CASAD service context matrix** as shown in table 4 is used to define the relationship between the contextual services and the contexts. This provides a detailed mapping of the services that are executed or provided after the occurrence of a context as well as the contexts that lead to the execution or the provision of a service. As an example, a warehouse worker can automatically be notified (service 1) in case of incoming goods (context 1) or the arrival of a new order (context 2).

Table 4. Basic structure of the CASAD service context matrix.

SCM ID	Context 1	Context 2	…
Service 1	X	X	…
…	…	…	…

The **CASAD context matrix** as shown in table 5 is used to define the relationship between the contexts, the contextual attributes, and the contextual values. This provides a detailed mapping of the elements that form each context and allows validating that each context is unique. During its creation, the contexts and contextual values are stated along the horizontal and vertical axis. In the intermediate cells, the contexts' contextual values are specified. The first field on the top left can be used to state a unique identifier.

Table 5. CASAD context matrix.

CM ID	Context 1	Context 2	Context 3	…
Contextual attribute 1	Contextual value 1/1		Contextual value 1/3	…
Contextual attribute 2	Contextual value 2/1	Contextual value 2/2		…
…	…	…	…	…

4.5. Additional matrices

In addition to the stated matrices, the format of the CASAD matrix can be used to document, analyze, and optimize a variety of context-related system properties. Among others, this includes software modules, hardware components, or interaction partners.

5. Analysis

A key advantage of the CASAD matrix method is the possibility to perform different evaluations. In particular, this includes the coordination, the clustering, and the validation of the context-related system properties. During the coordination, the elements are analysed, duplicates summarized, and incomplete descriptions completed.

Clustering rearranges the rows and columns with the goal of grouping similar elements and interaction patterns. For example, all contexts that consist of the same contextual attributes or that lead to the provision of the same services can be grouped. As noted by Eppinger and Browning [15], two contradictory goals must be weighed against each other. On the one hand, the number and the strength of interactions outside the clusters should be minimized. On the other hand, the cluster size should be minimized. Table 6 provides a theoretical example of a service context matrix after clustering. As shown, clustering normally leads to the accumulation of interactions around the diagonal axis. The width of this band around the axis is an indication for the system's modularity and should be as thin as possible.

Table 6. CASAD matrix analysis.

ID	Context 8	Context 2	Context 7	Context 3	…
Service 1	X	X			…
Service 9		X			…
Service 4			X	X	…
…	…	…	…	…	…

The validation includes the review of the elements with respect to different aspects. As an example, this can include the validation of the technical, economic, or organizational feasibility. In addition, the uniqueness of the contexts can be validated.

In addition to the stated evaluations, the CASAD matrix method is suitable for carrying out further assessments. Among others, this can include the optimization of the elements, the identification of indirect relationships, the identification of similarities between different context-aware systems, and the identification of the effects of system changes.

6. Practical application

In the following, the application of the CASAD matrix method is demonstrated in an industrial use case. Based on the EU Horizon 2020 research project FACTS4WORKERS [16], a context-aware information system for the shop floor of an injection molding company is developed. The system assists its users by automating work routines and by automatically providing the required digital instruction documents based on an understanding of their current activities. As specified by the CASAD process model of Rosenberger et al. [12], the system analysis is concerned with the definition and modelling of the work processes. Thereby, the contextually supported activities, the contextual services, and the contexts are determined. A detailed discussion of the corresponding UML models can be found in [17]. To improve clarity, ease understanding, and prevent the disclosure of company secrets, only a small excerpt of the use case is demonstrated. The work process discussed in this paper is concerned with the fault diagnosis and its resolution carried out by a machine operator.

6.1. Contextually supported activities

After production, the operator controls all parts visually to spot defects or imperfections. If no issues are detected, the parts are deburred and prepared for dispatch. In case of a faulty part, the machine operator defines the error, determines its cause, and fixes the problem if possible. Thereby, written and visual descriptions of the most common errors as well as the required skills for resolving the underlying issue assist the identification. If the error is not stated, the operator informs her/his supervisor who fixes it based on her/his knowledge. If the operator's skills are not sufficient, s/he also alerts the supervisor.

Several of the described tasks can benefit from a contextual assistance. First, the error determination is quite cumbersome due to the vast amount of produced parts and possible errors. Secondly, informing or alerting the supervisor is often difficult as they do not have a fixed workplace. Finally, assessing the required skills is not straight forward as they highly depend on the issue. Table 7 provides an overview over the contextually supported activities and indicates the benefits of a contextual assistance using a three-part scale (1: low, 2: medium, 3: high importance).

Table 7. Contextually supported activities.

Activity	Relevance
A.17: Determine production error	3
A.18: Inform supervisor	2
A.19: Check skills	1
A.20: Alert team leader	2

6.2. Contextual services

Based on the contextually supported activities, possible services are identified and analyzed. The activity service matrix (table 8) shows the contextual services in comparison to the contextually supported activities.

Table 8. Activity service matrix.

ASM-4	S.12-1	S.12-2	S.13	S.15
A.17	3	3		
A.18				2
A.19			1	
A.20				2

- **S.12-1 Automatic error detection:** The context-aware system automatically analyses each part to detect any errors.
- **S.12-2 Pre-sorted error descriptions:** The system provides a digitalized list of known production errors that is adapted to the currently produced product and ranked according to their probability of occurrence.
- **S.13 Automatic assessment:** The workers skills are assessed by the system based on the current issue and the operator's training level.
- **S.15 Automatic forwarding:** The supervisor is notified automatically about a missing error description or insufficient skills.

As shown, the activities A.18 and A.20 require the same kind of support, which reduces the development efforts. For supporting activity A.17, two different services can be implemented. The comparison shows that S.12-1 is characterized by a high degree of automation but poses major challenges. For example, the adaptation of the production machines can lead to warranty issues and the successful implementation cannot be guaranteed due to a lack of reference projects in the field of small batch production. Therefore, it is decided to only implement service S.12-2.

6.3. Contexts

Finally, the contexts are defined. The contextual attributes and values for each context are defined and documented using the context matrix (table 9). Thereby, the objective is to describe all relevant situations in a way a system can understand and reason upon. The current work process includes the following attributes:

- **Input:** This attribute includes the user input.
- **Location:** The location describes the user's current position.
- **Product:** This attribute identifies the product currently produced on a machine.
- **Issue:** This attribute describes the problem at hand.
- **Skills:** The skills describe a user's training level and are defined by number and topics of all successfully completed training courses.

Table 9. Activity context matrix.

CM-4	C.11	C.12	C.14	C.15
Input	D&S started	Missing error reported	D&S editor started	
Location	Machine ID#	Machine ID#		
Product	Product ID#	Product ID#		
Issue			Error ID#	Error ID#
Skills				Skills insufficient

- **C.11:** Context-class #11 is present whenever the user starts the program "D&S" to record an error. Within this class there are several contexts, each representing a different injection moulding machine as well as the product produced on this machine. Having this broad set of contexts allows the system to tailor the error descriptions specifically to the user's location and the product.

- **C.12:** Context-class #12 is present when the user reports a missing error description. Again, the class is subdivided into different contexts for each machine and product.
- **C.14:** Context-class #14 is present when the D&S editor is started to author a new error description. The class consists of a set of different contexts each representing a different error.
- **C.15:** Context-class #15 is present if the user's skills are not sufficient to tackle the issue. Again, the class is subdivided into different contexts for each error.

In addition to and aligned with the previous matrices, the activity context matrix (table 10) provides an overview over the contexts that identify each activity. As an example, activity A.17 (determine production error) is characterized by context C.11 (input, location, and product). The service context matrix (table 11) describes the contextual services that are executed after the occurrence of each context. As an example, service S.15 (automatic forwarding) is provided in the presence of context C.15 (issue, skills). Using these matrices provides a complementary perspective on the use case and therefore furthers comprehensibility.

Table 10. Activity context matrix.

ASM-4	C.11	C.12	C.14	C.15
A.17	X			
A.18			X	
A.19		X		
A.20				X

Table 11. Service context matrix.

ASM-4	C.11	C.12	C.14	C.15
S.12.2	X			
S.13		X		
S.15			X	X

7. Conclusion and outlook

The extended functionalities of industrial context-aware systems require the usage of adapted tools and techniques that reduce their complexity to a manageable level. This paper introduced the CASAD matrix method for documenting, analysing, and optimizing the properties of context-aware systems. Using this method helps system designers and developers to structure context-aware systems in a clear and understandable way. Further, the matrix reduces the system's complexity by mapping the relationships and correlations between its context-related components. Thereby, the matrix enables to state relevant information with different degrees of detail and to identify sub-systems, both leading to increased clarity. Having a generic basic structure ensures that the method is independent of the use case and application scenario.

The application of the CASAD matrix method has shown promising results in the industrial use case while structuring and analysing the context-related properties. Nevertheless, an extensive evaluation is still outstanding. Applying the method in a variety of use cases will allow making a general statement about its effectiveness and the advantages for system developers.

References

[1] Steward, D. V.: The design structure system: A method for managing the design of complex systems. In: IEEE transactions on Engineering Management, (3), pp. 71-74 (1981).

[2] Temdee, P., Prasad, R.: Context-aware communication and computing: Applications for smart environment. In: Springer International Publishing (2018).

[3] Perera, C., Zaslavsky, A., Christen, P., Georgakopoulos, D.: Context aware computing for the internet of things: A survey. In: IEEE communications surveys & tutorials, 16(1), pp. 414-454 (2013).

[4] Meyer, B. J., Drummond, T.: The Importance of Metric Learning for Robotic Vision: Open Set Recognition and Active Learning. arXiv preprint arXiv:1902.10363. (2019).

[5] Sparling, B.: Condition monitoring: Situational awareness brought to you by IoT. In: Trans-formers Magazine, 6(2), pp. 60-64 (2019).

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

[7] Zhou, K., Liu, T., Zhou, L.: Industry 4.0: Towards future industrial opportunities and chal-lenges. In: 12th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD), 2015, pp. 2147-2152. IEEE (2015).

[8] Wei, E. J., Chan, A. T.: Towards context-awareness in ubiquitous computing. In: Interna-tional Conference on Embedded and Ubiquitous Computing, pp. 706-717. Springer, Berlin, Heidelberg (2007).

[9] Rosenberger, P., Gerhard, D.: Context-awareness in industrial applications: definition, clas-sification and use case. In: Procedia CIRP, 72, pp. 1172-1177 (2018).

[10] Omasreiter, H., Metzker, E.: A context-driven use case creation process for specifying au-tomotive driver assistance systems. In: Proceedings of the 12th IEEE International Require-ments Engineering Conference, 2004, pp. 334-339. IEEE (2004).

[11] Choi, J., Lee, Y.: Use-case driven requirements analysis for context-aware systems. In: Computer Applications for Biotechnology, Multimedia, and Ubiquitous City. pp. 202-209. Springer, Berlin, Heidelberg (2012).

[12] Rosenberger, P., Gerhard, D., Rosenberger.: Context-Aware System Analysis: Introduction of a Process Model for Industrial Applications. In: Proceedings of the 20th International Conference on Enterprise Information Systems ICEIS, Vol. 2, pp. 368-375 (2018).

[13] Dey, A. K., Abowd, G. D., Salber, D.: A conceptual framework and a toolkit for supporting the rapid prototyping of context-aware applications. In: Human-computer interaction, 16(2), pp. 97-166 (2001).

[14] Greenberg, S.: Context as a dynamic construct. In: Human-Computer Interaction, 16(2), pp. 257-268. (2001).

[15] Eppinger, S. D., Browning, T. R.: Design structure matrix methods and applications. MIT press (2012).

[16] Facts4Workers: The Goals of FACTS4WORKERS, Facts4Workers. Available at: http://facts4workers.eu/index.php/the-goals/ (accessed at 12.06.2019).

[17] Rosenberger, P., Gerhard, D., Dumss, S.: Modelling the Behaviour of Context-aware Systems: State-of-the-Art Analysis and Introduction of a Customized UML Profile. In: Proceedings of the 7th International Conference on Model-Driven Engineering and Software Devel-opment, Vol. 1, pp. 521-528 (2019).