2015 48th Hawaii International Conference on System Sciences

MaXCept – Decision Support in Exception Handling through Unstructured Data Integration in the Production Context. An Integral Part of the Smart Factory.

Laura Bernadette Kassner
Graduate School advanced Manufacturing Engineering, Universität Stuttgart
laura.kassner@gsame.uni-stuttgart.de

Bernhard Mitschang
Institut für Parallele und Verteilte Systeme, Universität Stuttgart
Bernhard.mitschang@ipvs.uni-stuttgart.de

Abstract

Today, data from different sources and different phases of the product life cycle are usually analyzed in isolation and with considerable time delay. Real-time integrated analytics is especially beneficial in a production context. We present an architecture for data- and analytics-driven exception escalation in manufacturing and show the advantages of integrating unstructured data.

1. Introduction

Manufacturing companies are collecting large amounts of structured and unstructured data about products, processes and quality. This is facilitated by ongoing trends (cf. [21]) such as

- increased automation of production technology, especially sensor technology
- availability of fast connections for data transmission and large, affordable data storage
- easy access to computing devices, often mobile, for recording, organizing and analyzing data

Still, data from different sources and different phases of the product life cycle are usually analyzed in isolation and with considerable time delay [5,46]. The majority of unstructured data, which constitute 50 to 80% of data within an organization [34], are not accessed through analytics at all. Knowledge discovery is thus severely restricted and data-driven optimization of processes is conducted slowly if at all [12]. This means that complex, un-anticipated exceptions in real-time production which lead to flawed products or missed deadlines and thus to revenue loss may not be handled quickly and appropriately or may not even be discovered on time. With the technology and data available today, it is possible to amend this and develop analytics which greatly benefit the factories of the future. In this paper, we present MaXCept, a conceptual architecture for data- and analytics-driven **Manufacturing Exception Escalation** on and beyond the factory shop floor. We discuss in detail the data types, data sources, data needs and analytics potential for each component and each step of the exception escalation process. We present software components to be re-used or developed further in a prototypical implementation. The remainder of this paper is structured as follows: In ch. 2, we motivate unstructured data integration as a crucial step towards the smart factory of the future and present exception escalation as an important application scenario. In ch. 3, we give an overview of the MaXCept architecture and the exception escalation process which it supports. In ch. 4, we detail the data integration layer of MaXCept; in ch. 5, we discuss the individual components for exception escalation. In ch. 6 we review related work and existing components used for our prototypical implementation; ch.7 contains conclusions and future directions.

2. Motivation

To motivate our architecture, we address developments towards a smart shop floor environment (2.1.) and present contexts for unstructured data analytics in exception handling (2.2.). We also introduce the case study we use as a working example (2.3.).

2.1. The Smart Factory and the Human Factor

Production today is under pressure to become even more flexible and easily adaptable, due to global megatrends [21] such as more complex products, higher demand for customization and faster, more global markets. The development of manufacturing towards more automation, knowledge-driven and data-driven decentralized planning and the incorporation of smart technologies has been described as the *fourth industrial revolution* [21]. At its core lies the concept of the *smart factory* [50], characterized by

- cyber-physical production systems "capable of autonomously exchanging information, triggering actions and controlling each other independently" [21]
- flexible, de-centralized task assignment
- real-time analytics and feedback loops
- a high degree of automation

Unlike its predecessors in the era of computer-integrated manufacturing (CIM), the smart factory is not intended to be empty of human workers. Instead, uniquely human skills will become more important. Today, human workers spend much of their time performing simple, repetitive manual tasks. With increased automation, human workers move on to tasks which tax their uniquely human skills such as flexible and creative problem-solving and decision-making [21,41]. They will always excel over machines at making decisions under uncertain and unpredictable conditions [1]. The factory is a very data-rich environment, and the information available on the shop floor will drastically increase in the near future. In the office environment, this has already happened and is subject to research (e.g. [31,42]). Additionally, the factory requires physical tasks to be performed on a tight schedule, thus positing an unprecedented challenge. Sophisticated filtering and assistant systems are therefore needed to maintain productivity and innovation in a fast-paced, flexibly changing and complex environment and to enable the human worker to make optimal decisions.

2.2. Exception Handling with Unstructured Data

We define an *exception* in the production context as any deviation from the production plan (similar to [24]), either directly through failure to fulfill a process step or indirectly through minor deviations which add up to a failure to complete a process under optimal conditions, e.g. a delay in time or a sloppily manufactured work piece. Humans communicate information in the form of unstructured data, e.g. written failure reports. Thus, most knowledge about flexible exception solution on the shop floor is at present available in unstructured form. To ensure that exceptions are discovered and solved in a timely fashion, we need to use both structured data – machine messages, historical errors, workflows – and unstructured data – image, audio and video documentation of known or discovered problems, text descriptions of best practices for solutions. The exception handling IT infrastructure which we describe can be realized in (1) real-time, online analytics and problem solving in a smart, connected shop floor context with immediate automated feedback for the smart factories of the near future and in (2) exception escalation in production-near contexts such as remote machine maintenance and off-site customer support, which does not require immediate feedback and can be implemented with the tools and infrastructure of today.

2.3. Case Study

Company SUP is a supplier of motor parts for car motors, producing gears and cogs in factory FC. Company MACH is the provider of machines for this factory, in particular, a cutting machine CM, a transfer robot CR and a quality control station CQ involved in a short process with three steps: (1) cog cutting, (2) transfer to a storage area and (3) subsequent quality scans. The following workers are involved in the production process and machine support respectively:

- A (SUP) – worker in cog quality control CQ at the storage area, listed as quality expert in the SUP social network
- B (SUP) – worker at cog cutting machine CM
- C (SUP) – surveillance worker at transfer robot, formerly worked at machine CM and contributed a tutorial on how to calibrate the tool for new work pieces
- E (MACH) – worker in technical support for machines CM and CR
- F (MACH) – informally interested in robot CR, has contributed to troubleshooting and documentation in intranet forums

Information about the manufacturing process structure and past process instances is stored in structured form, documentation for the machines and tutorials exist in unstructured form. The two companies have intranets with user profiles and forums for discussion and troubleshooting.

3. An Architecture for Manufacturing Exception Escalation (MaXCept)

In this chapter, we present MaXCept, an architecture for decision support through automated exception escalation. We give a general overview of the architecture in 3.1. and describe the exception escalation process in 3.2.

3.1. Architecture Overview

The MaXCept architecture, shown in Fig.1, consists of four layers:

(1) The *execution layer*, which is equivalent to the shop floor environment – containing machines / CPPS with their controlling units and sensors; human workers with their scheduled tasks and tool equipment; and smart devices with sensors and apps.

(2) The *integration layer*, where data from different sources and product life cycle phases are integrated into a knowledge repository as well as into a computational model of the shop floor. The knowledge repository also houses all processes runnable in the factory and corresponding workflow representations.

(3) The *analytics layer*, which contains the central IT components necessary for the exception escalation process, as well as other analytics components, scheduling / managing and notification / communication tooling.

(4) The *presentation layer,* which provides the human user with access to analytics results, communication tools and exception notifications. The user interface can take the shape of an app on a mobile device, a user terminal connecting to a machine or cyber-physical unit, or a wearable / augmented reality component.

3.2 Exception Escalation Process

The exception escalation process has three core phases, with four associated solution and matching tasks (cf. Fig. 2). The **core phases** are the following:

(1) *Exception Recognition*: Exceptions are automatically recognized due to explicit alerts or as the result of data mining on integrated manufacturing data. Known exceptions designated by a unique error code may be *resolved automatically* without human decision-making.

(2) *Exception Classification*: Exceptions are automatically assigned a number of features in preparation for solution discovery.

(3) *Exception Escalation*: Based on these features, one of the following three steps is taken:

- *Automatic solution* if the features of the exception match up with a known exception
- *Solution recommendation* if the exception resembles several known exceptions.

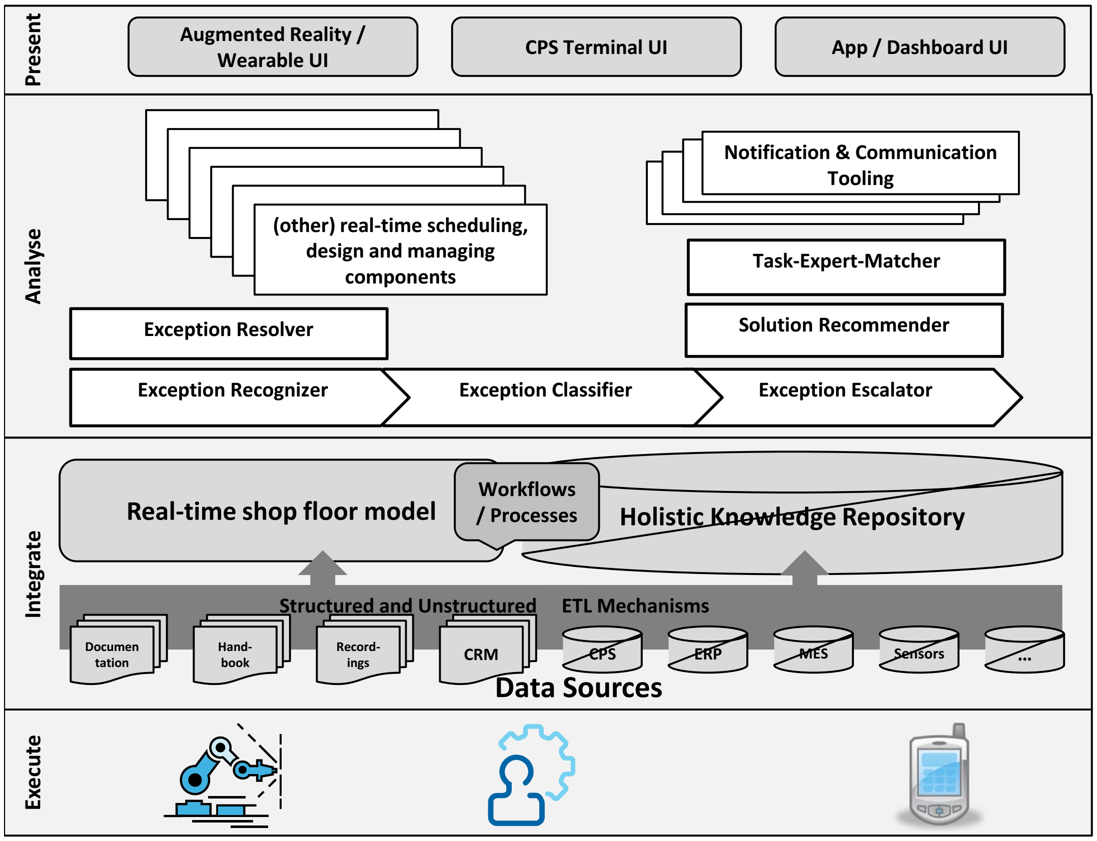

Figure 1. MaXCept architecture for Manufacturing Exception Escalation

- If no solution can be recommended, *experts* on the topic of the exception *are determined* through a matching process.

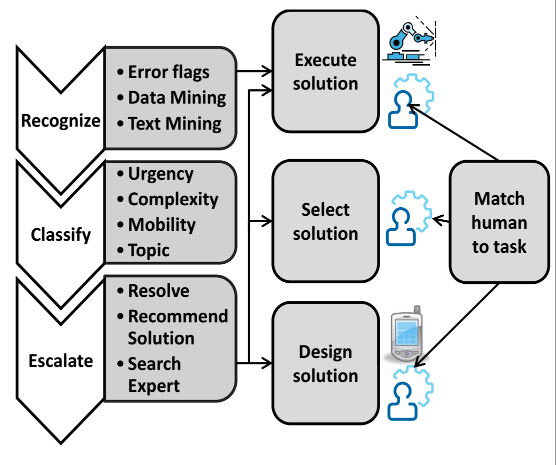

Figure 2. Exception Escalation Process

The associated **solution and matching tasks** are initialized at different steps in the process depending on the type of exception:

(1) *Solution Execution* can be started either after error recognition, after recommendation-based solution selection, or after solution design. It is handled by the exception resolver component, which assigns process steps to machines (e.g. automatically exchanging a worn-out tool) and human workers (e.g. fetching spare parts or conducting complex repairs).

(2) *Solution Selection* is initialized when several potential solutions are automatically recommended. It must be carried out by a human worker.

(3) *Solution Design* is evoked when no solution could be recommended. This task is carried out by a human.

(4) Wherever humans are involved, the *task and the human participant must be matched to each other* based on appropriate parameters.

4. Data Integration

The data integration layer comprises data sources, integration components and warehousing and modeling components. Data sources are diverse and contain structured and unstructured data. **Structured data** are produced by *sensors* on the shop floor, which record for instance temperature, humidity, tool abrasion, or location of mobile components. *Manufacturing execution systems (MES)* and *enterprise resource planning systems (ERP)* as well as *machine control systems* also contribute structured data. **Unstructured data** originating on the shop floor are typically produced by humans. They mainly include *error documentation* in text format as well as possibly *voice or noise recordings*, *photographs or video recordings* of failing machines or flawed work pieces and components. For the purpose of our research, we focus on unstructured text data. *Tutorials and handbooks* for human workers also come in unstructured form. Other unstructured data sources from outside the production phase are complaints in *customer relationship management (CRM)* systems or reviews from *social media*. The **holistic knowledge repository** is a data warehouse integrating structured and unstructured data as well as analytics results, e.g. *mining models* for error detection or solution recommendation and *worker expertise data*. We base the design of the knowledge repository on the concept put forth by [14]. To be integrated into the knowledge repository, data must be preprocessed – structured data with *standard Extract-Transform-Load (ETL)* techniques, unstructured data with *generic natural language processing (NLP)* to be enriched with structure (cf. [Author, 2014]). In the course of analytical processing, semantic links between structured and unstructured data are established and extracted information is stored in new configurations.

The **real-time shop floor model** contains representations of

- stationary objects such as machines, rooms or buildings
- mobile objects such as vehicles, robots and human workers
- environment variables such as temperature, humidity or velocity
- tasks and assigned agents, events and messages

The model is similar in concept to the one developed by [16]. We also adopt their distinction of *static* and *dynamic data*.

5. Decision Support Components for Exception Escalation

In this chapter, we describe the architecture components of the exception escalation process. We discuss the exception recognizer in 5.1., the exception classifier in 5.2., the exception escalator in 5.3. and the solution recommender and task-expert matcher in 5.4. and 5.5. Application examples are provided from the context of the case study.

5.1 Exception Recognizer

The exception recognizer component deals with the discovery of exceptions (Fig. 3). Exception discoverability is based on two feature dimensions: *knownness* and *complexity*. An exception is known if it

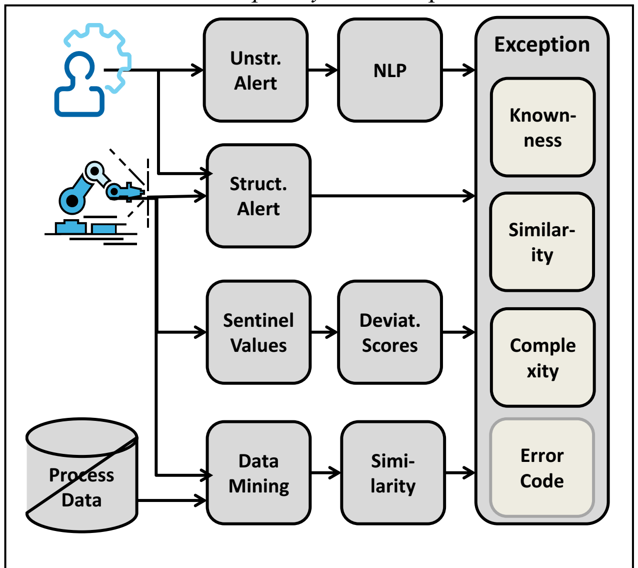

Figure 3. Exception Recognition

has been anticipated during process design or has occurred before. Its complexity depends on the number of process steps involved. An exception which is *known* and *simple* can easily be recognized via sensor technology, e.g. when the cutting tool of machine CM must be exchanged because of abrasion. Typically, these exceptions also have straightforward solution processes retrievable by an unique error code, which can then be started automatically. An exception can also be *known* but *complex*, e.g. production steps are slightly delayed at machine CM because of permissible tool abrasion and at the quality control station CQ because of an inexperienced worker, such that an overall delay beyond the tolerance limit for timely production must be expected. To recognize this sort of exception, we require more sophisticated recognition mechanisms, e.g. computing a deviation score from several sentinel values or recognizing equivalence to previously seen exceptions through predictive analytics on historical process data (for an example with structured process data only, see [13]). These exceptions may also have straightforward solution processes that can be selected and started automatically. If an exception is *unknown* because it has neither occurred before nor been anticipated during process design – i.e. a true exception according to [35] - , it may still be automatically *discoverable* through data mining on real-time process data without pre-selected sentinel values. For example, jagged cog edges can be measured and recognized as deviant from previously seen edges during quality control without a predefined jaggedness threshold. Today, sensor technology is not as ubiquitous as it will be in the smart factories of the future, and manufacturers are still successfully running machines built several decades ago. This means that the human observer will play a large role in detecting exceptions on the shop floor for a while, and that alerts sent to exception recognition will be in the form of unstructured data.

The tasks of the exception recognizer component thus encompass the following (cf. Fig. 3):

- Catching structured alerts thrown by humans or machines for known and simple exceptions
- Decoding unstructured alerts thrown by humans
- Watching sentinel values and computing deviation scores for known and complex exceptions
- Watching processes and applying data mining to compare them to historical data to detect known and unknown complex exceptions
- Assigning error code (if applicable) and knownness, similarity and complexity features to the discovered exception

5.2. Exception Classifier

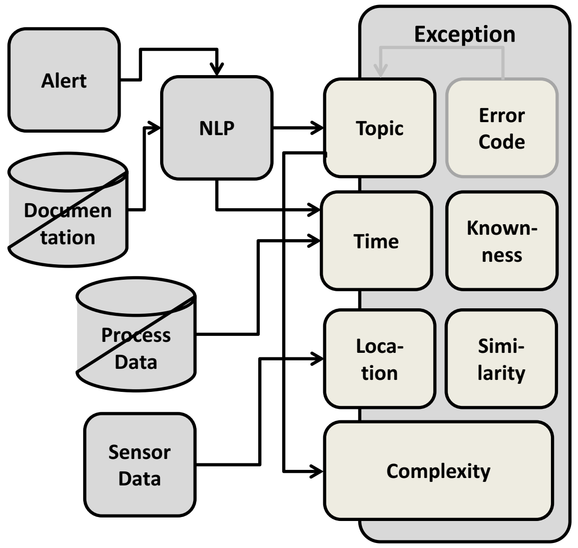

Figure 4. Assigning Additional Features

In order to recommend solutions for complex and possibly unknown exceptions, several additional features are of interest. They can be clustered into four categories:

(1) *Topic*: Which machines, machine parts, work pieces or workers are affected in which way? This information can be extracted from unstructured exception alerts, such as worker C sending an alert message via smartphone "transfer to storage area stopped because robot arm CR is stuck in palette". Comparison of the

detected topic to historical topics can also lead to the assignment of an available error code for retrieving a known solution.

(2) *Time*: Does the exception need to be resolved immediately or within a specific time window? This can be determined by consulting records for known exceptions or considering signal words and expressions in the text (“cutting tool at machine CM is broken and needs to be exchanged *immediately*”).

(3) *Location*: Is the exact location of the exception known? Are several locations involved? Is there remote access to the affected components (e.g. for software problems) or do they need to be treated on-site (physical repairs)?

(4) *Complexity*: Further complexity measures can be derived from the topic features: the affected components give clues as to whether special training is needed to carry out the solution.

These features are assigned by the exception classifier component with the help of data mining and text mining (Fig. 4).

5.3. Exception Escalator

The exception escalator component uses the exception features to decide how the exception is to be solved. This automatic decision process uses hand-crafted or data-mined decision models retrieved from the knowledge repository and has three possible outcomes, mentioned already in 3.2:

(1) Automatic assignment and triggering of a solution process
(2) Recommendation of several potential solutions to a human worker
(3) Alerting a human expert to the exception who will then design a solution process.

5.4. Solution Recommender

The solution recommender draws on the following data:

- Similarity measures of the current exception to previously seen ones, for example, surface roughness or jag size in damaged work pieces, temperatures at the failing machine, etc.
- Exception features determined in the classifier
- Solution processes assigned to known and complex exceptions
- Atomic tasks, such as solution steps assigned to known and simple exceptions

There are two strategies which the recommender can follow, based on data availability and exception features (illustrated in Fig. 5):

(1) Recommending existing solutions based on similarity to other exceptions
(2) Tailoring new solution processes from classifier features

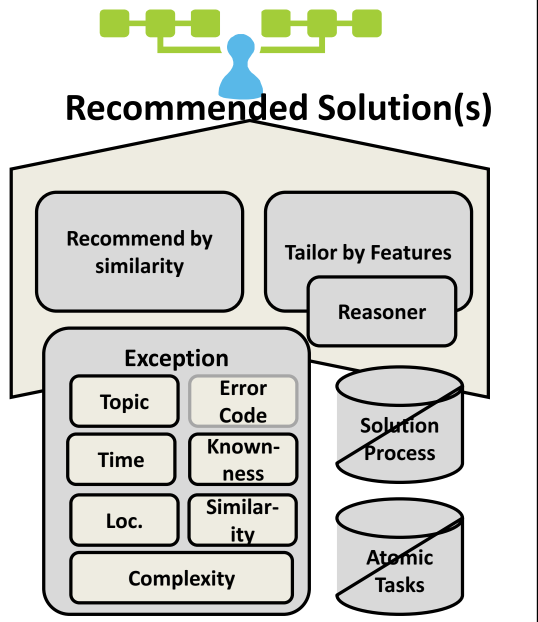

Figure 5. Solution Recommendation

The first strategy is straightforward: The solutions associated with similar exceptions can be ranked according to the similarity scores. For example, delays in the current process execution may be similar to several historical examples, one of which was solved by substituting an experienced worker at the quality control station and one of which was solved by simply speeding up the cutting machine. There will also be cases where similarity is low, low-confidence, non-existent. In such cases, the second strategy is pursued: Based on topic features such as *affected machine part* (e.g. “robot arm”) and *nature of failure* (e.g. “jammed”), a reasoner using domain-specific concept hierarchies and relations can recommend a series of atomic tasks.

5.5. Task-Expert Matcher

The matching of the right human worker to a particular task is necessary at several points in the exception escalation and solution process. The task-expert matcher component (Fig. 6) makes use of

(1) Exception features
(2) Task features of solution step or decision/design tasks
(3) Features of human workers, such as areas of expertise, current location, availability (e.g. current task, scheduled breaks) and

reachability (e.g. via text message, phone call, or video conference)

We have discussed (1) and will now shed some light on how to collect and organize (2) and (3). **Solution tasks** can be assigned *topic, time, location* and *complexity* features, which will differ from the associated exception features. For example, solving a complex exception such as the delay caused at two interdependent machines by inexactly calibrated robot arm motions at CR and a highly eroded tool at CM can involve *simple steps* requiring *high mobility* and *little expertise*, such as fetching a new tool from a designated storage area, which can be carried out by worker B, and *complex steps* requiring *no mobility* but *high expertise*, such as re-calibrating the robot arm via remote access, which can be carried out by support technician E. **Decision and design tasks** are marked as high-complexity and assigned topic, time and location features in accordance with the exception and potentially the set of solutions under investigation. The **features of human workers** which we need to know in order to match them to tasks split into *dynamic* and near-*static* features: Location, current task, availability and reachability are subject to quick changes, whereas areas of expertise are stable over time. The dynamic data are present in the real-time shop floor model; their original data sources range from work assignment schedules within ERP systems and login information from stationary machine terminals to location and reachability data transmitted by smart devices. These are all structured data sources, whereas the near-static expertise data is derived from structured and unstructured data sources:

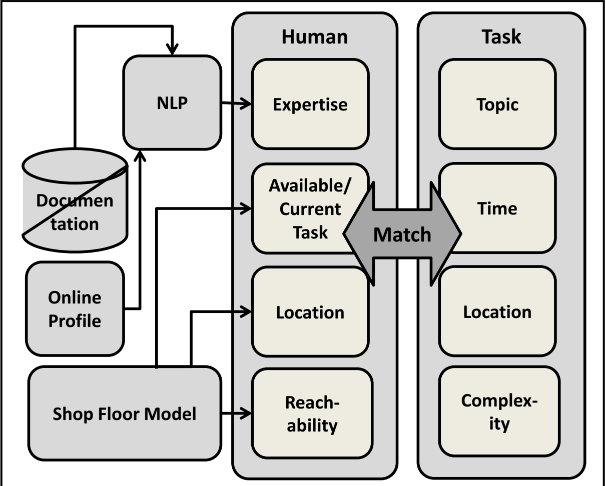

Figure 6. Task-Expert Matching

- Employees can assign themselves topics of expertise and interest in *company-internal social networks*, and have them endorsed by others to create a confidence score.
- *Authorship of documentation and tutorial texts* on certain topics also indicates expertise – for example, worker F would be found as a candidate for recalibrating the robot arm based on troubleshooting documentation authorship.
- Unstructured descriptions and structured metadata on *tasks an employee has previously accomplished* can be mined to derive competence profiles, e.g. worker C's experience with machine CM can be retrieved from past work schedules.
- Finally, an employee can be the *designated contact person* for a specific task or the *explicit requester of information*, e.g. worker C sending an exception alert about robot CR and wanting to retrieve a solution.

Once the best fit between a task and a worker has been determined, the worker is informed of the assignment and given the necessary information for fulfilling it via one of the available media.

6. Towards an Implementation

In this chapter, we discuss the technology needed for an implementation of the Manufacturing Exception Escalation system. We first point out central challenges in 6.1., then discuss related work and existing components in 6.2.

6.1 Central Challenges

The main challenge is the integration of very diverse data which are structured and unstructured, frequently noisy and underspecified, into the knowledge repository as well as into the shop floor model. While the integration of legacy data into the knowledge repository and the data mining for potential solutions of known historical problems can happen *off-line*, the integration of data into the shop floor model and the exception escalation pipeline have to run in or near *real-time.* This is slightly more difficult for unstructured data because there is more preprocessing involved. NLP components have to robustly process text and speech data which are dissimilar to the standard written language usually employed to train statistical NLP models. Recent work on real-time natural language processing and question answering [7] makes a valuable contribution towards solving this problem.

6.2 Existing Components and Related Work

Approaches towards the realization of a **smart, connected factory** can be found e.g. in [39], an implementation of an engineering service bus for data and service integration on the shop floor which has been realized in a small-scale model of a smart factory. The design and a proof of implementation for a **manufacturing knowledge repository** have been put forth by [14], which provides a detailed meta model for manufacturing processes and a model for structured and unstructured insights which can be used for process optimization. While the focus is not on real-time exception handling but on predictive analytics and while the meta model only makes mention of exceptions ('failures') but does not attempt to resolve them, the structure of the knowledge repository still provides a very useful basis for the corresponding component in the MaXCept architecture. We are currently researching methods for the generation and maintenance of semantic links between structured and unstructured insights within the knowledge repository for a use case from the automotive industry [Author, 2014]; these links will be of central importance for data mining, solution recommendation and task-expert matching. Both [14,39] constitute the basis for components in our implementation of the MaXCept architecture.

[47,48,49] have developed methods to **integrate context information** – e.g. from sensors – into a real-time **workflow model** of processes in a shop floor environment. This work may be taken as a starting point for the implementation of the MaXCept shop floor model, but must be extended to support unstructured data sources as input to the workflow model. [30] is conceptually valuable for the development of MaXCept because it computes **similarities of processes, subprocesses and activities** to optimize processes at design time by suggesting appropriate process step sequences. These similarity measures are mainly based on the linguistic labels of activities, but comparable similarity measures can be computed with the features of tasks, exceptions and persons which we discussed in this paper. They can then be used at run time in the MaXCept solution recommendation component.

[29] treat **automated process step decisions** on the basis of integrated process and operational data as a classification problem and successfully test this approach on structured integrated data. While they remark that unstructured data are also relevant, they do not address the question of how to integrate them, and the setting is not a factory context with real-time requirements. We will investigate in how far the real-time solution recommendation task can also be treated as a classification problem and what the impact of including unstructured data sources will be.

[17] **integrates collaboration tools for decision processes with workflow systems**, addressing the need to "identify situations where formalized solutions do not exist" in order to contact humans who will cooperate to informally solve the problem but providing no clear strategy to do so. It constitutes a relevant starting point for designing the **user interface** of the MaXCept architecture, as do [11,15] which have developed a worker information and communication dashboard (in the form of a mobile app) that may be integrated into our implementation.

[1] addresses **exception handling in business process management** with a focus on general exception types and task types which we have taken as a starting point to develop our notions of exception discoverability and exception and task features. [26] develop a case-based reasoning architecture for exception handling in a workflow system which may serve as a basis for implementing the exception handling pipeline in MaXCept but needs to be adapted to the new context of the integrated, data-rich factory environment.

For the processing and integration of unstructured text data, both generic and domain-specific **NLP tools** and resources will be necessary. For generic NLP, standard pipeline components such as the Stanford Core NLP package [9,45] or OpenNLP [2] can be used; domain-specific components will be developed on this basis. The Apache UIMA standard [8] is adhered to throughout the implementation of MaXCept, especially within the data analytics components. It offers integration of most standard NLP components, it is modular and has high scale-out capabilities which make it ideal for real-time analytics.

Topic identification is a major focus in exception escalation with unstructured data. We predict that a simple statistical approach will handle the noisy non-standard data we are confronted with on the shop floor slightly better than a sophisticated rule-based approach. Topics can be organized and clustered with the help of **taxonomies** or **ontologies**. Examples of generic world-knowledge ontologies include WordNet for English [6] and GermaNet for German [19]. For the shop floor environment, domain-specific ontologies will need to be used. Examples of manufacturing-specific ontologies exist for design [40] and assembly [10]. While ontologies and taxonomies can be costly if they need to be hand-crafted, there also exist approaches to automating their generation [4,23,32,44]. Initial experimentation in research and real-life industry settings suggest that a shallow custom taxonomy (such as used in [36]) is sufficient for classification and topic markup tasks. Since the shop floor is a "hands-busy, eyes-busy, and mobility required situation[…]" [38], speech-to-text components such as described in [43] and topic identification on spoken text [27] are also of interest.

Data-driven **expert identification** has already been addressed in [22], which integrates structured and unstructured data from various internet resources into an expert search portal with sophisticated focused search – on the basis of taxonomy-like keyword graphs – and pattern recognition for person and expertise identification. [20] presents an interactive, self-improving expert search mechanism with natural language interfaces which is integrated into a social network and finds experts to answer users' questions on a wide range of topics based on user-created texts, explicitly specified interests/areas of expertise, and user-to-user recommendations. [25] compare a range of similarity measures for matching up expert profiles with each other; these can also be used to match expert profiles to task profiles. These and similar approaches can be applied to data sources from the manufacturing context such as company-internal social networks, handbooks and tutorials with authorship information, and insights from the knowledge repository associated with their creators as in [14]. Lacking from these approaches so far is the integration of real-time context data, which we will include in our development of MaXCept.

Choosing optimal modalities for information presentation, teamwork, innovation and communication has been well researched for the office context, e.g. [3,28,37,42]. In particular, decision support for the context-sensitive choice of communication medium such as [33] may be of relevance for communicating task assignments.

Data mining is crucial in several components of the MaXCept architecture: It will be needed for the recognition of exceptions from raw data, for the classification of exceptions according to the features we listed, and for the discovery of potential solutions and experts. We will use a freely available toolkit such as [18], which has been successfully employed throughout the scientific community.

7. Conclusions and Outlook

We have shown that exception handling in manufacturing, especially real-time exception escalation on the factory shop floor, can greatly benefit from data-driven analytics tools for decision support throughout the process of exception escalation, and that the integration of unstructured data is crucial for tapping into existing knowledge about solution processes. We have described an architecture for supporting exception escalation with comprehensive data integration, discussed each component and process step in detail and shown that there exists a rich research context for all technologies relevant for its implementation. After finalizing the protoype of the MaXCept architecture, we will proceed to evaluating it in a realistic smart factory context.

8. References

[1] Antunes, P. BPM and Exception Handling : Focus on Organizational Resilience. *41*, 3 (2011), 383–392.

[2] Baldridge, J. and Morton, T. OpenNLP. 2004.

[3] Bhagwatwar, A., Massey, A., and Dennis, A.R. Creative Virtual Environments: Effect of Supraliminal Priming on Team Brainstorming. *2013 46th Hawaii International Conference on System Sciences*, (2013), 215–224.

[4] Biemann, C. Ontology Learning from Text : A Survey of Methods. *20*, 2 (2005), 75–93.

[5] Blumberg, R. and Atre, S. The problem with unstructured data. *DM REVIEW*, (2003), 42–45.

[6] Fellbaum, C. *WordNet*. Blackwell Publishing Ltd, 1999.

[7] Ferrucci, D., Brown, E., Chu-Carroll, J., and Fan, J. Building Watson: An overview of the DeepQA project. *AI magazine*, (2010), 59–79.

[8] Ferrucci, D. and Lally, A. UIMA: an architectural approach to unstructured information processing in the corporate research environment. *Natural Language Engineering 10*, 3-4 (2004), 327–348.

[9] Finkel, J.R., Grenager, T., and Manning, C. Incorporating Non-local Information into Information Extraction Systems by Gibbs Sampling. *Proceedings of the 43nd Annual Meeting of the Association for Computational Linguistics (ACL 2005)*, (2005), 363–370.

[10] Fiorentini, X., Gambino, I., Liang, V., and Rachuri, S. An ontology for assembly representation. (2007).

[11] Gröger, C., Hillmann, M., Hahn, F., Mitschang, B., and Westkämper, E. The Operational Process Dashboard for Manufacturing. *46*, (2013), 0–5.

[12] Gröger, C., Niedermann, F., Schwarz, H., and Mitschang, B. Supporting Manufacturing Design by Analytics. *Proceedings of the 2012 IEEE 16th International Conference on Computer Supported Cooperative Work in Design, CSCWD 2012 (2012)*, (2012).

[13] Gröger, C., Schwarz, H., and Mitschang, B. Prescriptive Analytics for Recommendation-Based Business Process Optimization. 3.

[14] Gröger, C., Schwarz, H., and Mitschang, B. The Manufacturing Knowledge Repository. Consolidating Knowledge to Enable Holistic Process Knowledge Management in Manufacturing. *Proceedings of the 16th International Conference on Enterprise IInformation Systems (ICEIS), 27-30 April, 2014, Lisbon, Portugal.*, (2014).

[15] Gröger, C. and Stach, C. The Mobile Manufacturing Dashboard. 1–3.

[16] Grossmann, M., Bauer, M., Hönle, N., Käppeler, U., Nicklas, D., and Schwarz, T. Efficiently Managing Context Information for Large-scale Scenarios. PerCom (2005).

[17] Guimarães, N., Antunes, P., and Pereira, A. The integration of workflow systems and collaboration tools. *Workflow Management Systems ... 164*, (1998), 222–245.

[18] Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., and Witten, I.H. The WEKA Data Mining Software: An Update. *SIGKDD Explorations 11*, 1 (2009), 10–18.

[19] Hamp, B. and Feldweg, H. Germanet - a lexical-semantic net for german. *Proceedings of ACL workshop Automatic Information Extraction and Building of Lexical Semantic Resources for NLP Applications.*, (1997).
[20] Horowitz, D. and Kamvar, S.D. The anatomy of a large-scale social search engine. *Proceedings of the 19th international conference on World wide web - WWW '10*, (2010), 431.
[21] Kagermann, H., Wahlster, W., and Helbig, J. Umsetzungsempfehlungen für das Zukunftsprojekt Industrie 4.0. *Abschlussbericht des Arbeitskreises Industrie*, April (2012).
[22] Kaiser, F., Schimpf, S., Schwarz, H., Jakob, M., and Beucker, S. *Internetgestützte Expertenidentifikation zur Unterstützung der frühen Innovationsphasen*. 2007.
[23] Kassner, L., Nastase, V., and Strube, M. Acquiring a Taxonomy from the German Wikipedia. 2143–2146.
[24] Klein, M. and Dellarocas, C. A Knowledge-based Approach to Handling Exceptions in Workflow Systems. *Computer Supported Cooperative Work (CSCW) 9*, 3-4 (2000), 399–412.
[25] Kraß, W. and Försterling, U. Evaluation of similarity measures for knowledge profiles from an expert directory. *Proceedings of the 12th International Conference on Knowledge Management and Knowledge Technologies - i-KNOW '12*, (2012), 1.
[26] Luo, Z., Sheth, A., Kochut, K., and Miller, J. Exception handling in workflow systems. *Applied Intelligence*, (2000), 125–147.
[27] McDonough, J., Ng, K., Gish, H., and Rohlicek, J.R. Approaches to topic identification on the switchboard corpus. *… , Speech, and Signal …*, (1994), 385–388.
[28] Mennecke, B.E. The Effects of Media and Task on User Performance : A Test of the Task-Media Fit Hypothesis. Panko 1992 (2000), 507–529.
[29] Niedermann, F. and Maier, B. Automated process decision making based on integrated source data. *Business Information …*, (2011).
[30] Niedermann, F., Radeschutz, S., and Mitschang, B. Design-Time Process Optimization through Optimization Patterns and Process Model Matching. *2010 IEEE 12th Conference on Commerce and Enterprise Computing*, (2010), 48–55.
[31] Van Osch, W. and Coursaris, C.K. Organizational Social Media: A Comprehensive Framework and Research Agenda. *2013 46th Hawaii International Conference on System Sciences*, (2013), 700–707.
[32] Pivk, A. Automatic ontology generation from Web tabular structures. *19*, (2006), 83–85.
[33] Reiter, M., Houy, C., Fettke, P., and Loos, P. Context-Sensitive Collaboration in Service Processes through the Integration of Telecommunication Technology and Business Process Management. *2013 46th Hawaii International Conference on System Sciences*, (2013), 491–500.
[34] Russom, P. BI Search and Text Analytics. *TDWI Best Practices Report*, (2007).
[35] Saastamoinen, H. and White, G. On handling exceptions. *Proceedings of conference on …*, (1995), 302–310.
[36] Schierle, M. and Trabold, D. Extraction of Failure Graphs from Structured and Unstructured Data. *2008 Seventh International Conference on Machine Learning and Applications*, (2008), 324–330.
[37] Seeber, I., Waldhart, G., Maier, R., Hecht, M., Kaschig, A., and Hrastnik, J. Better Together: Exploring the Effects of Knowledge Application, Support for Innovation and Team Characteristics on Team Performance. *2013 46th Hawaii International Conference on System Sciences*, (2013), 559–568.
[38] Shneiderman, B. The limits of speech recognition. *Communications of the ACM 43*, 9 (2000), 63–65.
[39] Silcher, S., Konigsberger, J., Reimann, P., and Mitschang, B. Cooperative service registries for the service-based Product Lifecycle Management architecture. *Proceedings of the 2013 IEEE 17th International Conference on Computer Supported Cooperative Work in Design (CSCWD)*, (2013), 439–446.
[40] Sim, S.K. and Duffy, A.H.B. Towards an ontology of generic engineering design activities. *Research in Engineering Design 14*, 4 (2003), 200–223.
[41] Spath, D., Ganschar, O., and Gerlach, S. *Produktionsarbeit der Zukunft - Industrie 4.0*. Fraunhofer Verlag, Stuttgart, 2013.
[42] Standaert, W., Muylle, S., and Basu, a. Assessing the effectiveness of telepresence for business meetings. *2013 46th Hawaii International Conference on System Sciences*, (2013), 549–558.
[43] Stolcke, A., Member, S., Chen, B., et al. Recent Innovations in Speech-to-Text Transcription at SRI-ICSI-UW. *14*, 5 (2006), 1729–1744.
[44] Tho, Q.T., Hui, S.C., Member, S., Fong, A.C.M., and Quan, T.T. Automatic Fuzzy Ontology Generation for Semantic Web. *18*, 3 (2006), 842–856.
[45] Toutanova, K. and Manning, C.D. Enriching the Knowledge Sources Used in a Maximum Entropy Part-of-Speech Tagger. *Proceedings of the Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*, (2000).
[46] Unit, E.I. *Knowledge management in manufacturing*. 2007.
[47] Wieland, M., Kaczmarczyk, P., and Nicklas, D. Context Integration for Smart Workflows. *2008 Sixth Annual IEEE International Conference on Pervasive Computing and Communications (PerCom)*, (2008), 239–242.
[48] Wieland, M., Kopp, O., Nicklas, D., and Leymann, F. Towards context-aware workflows. *CAiSE*, (2007), 1–15.
[49] Wieland, M., Leymann, F., and Jendoubi, L. Task-orientierte Anwendungen in einer Smart Factory. *MMS*, (2006).
[50] Zuehlke, D. SmartFactory—Towards a factory-of-things. *Annual Reviews in Control 34*, 1 (2010), 129–138.