Whitepaper: AI engineering by means of a functional architecture – From the identification of benefits to an experimental feasibility study

A. Friedmann[(1)], V. Mees[(1)]

Introduction

The research project ProKInect focuses on collaborative condition monitoring with distributed AI agents. It addresses the cross-company development and use of condition monitoring systems for production facilities. Currently, the implementation and use of such a system is hindered by concerns regarding the confidentiality and sovereignty of shared data, as well as fears regarding the loss of the integrity of own EDP system and the data contained. One potential solution to these challenges was identified and tested in ProKInect. It is presented in the ProKInect white paper, "Multilateral Data Sharing in Secure Data Spaces." Additionally, cross-stakeholder development presents a challenge. This white paper outlines the steps taken in ProKInect in a sequence that demonstrates how this challenge can be met.

Description of the working method

General mode of operation

The general working method is consistent with the usual procedure for cross-company project processing. It should be noted, however, that in this case there is a significantly higher number of links between the working groups compared to "classic" projects. Furthermore, the individual work steps between the coordination meetings are relatively small, demands for closely interwoven activities.

A method of working that involves an intensive exchange between all partners, moderated by one of the partners, has proven to be an effective approach. This type of collaboration is conducted in one-hour virtual meetings, during which representatives from all partner organizations participate. The meetings are held every one to two weeks. The meetings are primarily used to compare the statuses of the last activities and to determine the next tasks. As a result, each meeting concludes with a written list of upcoming tasks prepared by the moderator. The review of this task list then serves as an introduction to the next meeting. At specific points (see below), virtual meetings are complemented by face-to-face meetings. The coordination and cooperative work agreed upon in the regular meetings is carried out in meetings organized independently by the respective parties between regular meetings. The schedule of these meetings is dependent on the task at hand.

Preparation

Prior to starting into the development process, it is vital to establish a shared agreement about the manner of collaboration and to set up a list of the inputs available from previous phases. As part of the organizational coordination, it is essential to determine the members of the development team, the role of the moderator, and the frequency of virtual status meetings. To fulfill the role of moderator effectively in the development context described here, it is important that the moderator does not have to make any content-related contributions. This approach avoids any potential conflicts of roles that could result in delays to the tightly scheduled development process.

An online whiteboard is used to collect inputs from all partners simultaneously, which then is utilized in both online meetings and face-to-face events. The collected inputs are presented to all

[1] andreas.friedmann@lbf.fraunhfoer.de, valentin.mees@lbf.fraunhofer.de, Fraunhofer LBF, Darmstadt

participants, questions are answered, and, where necessary, missing inputs are obtained. Subsequently, a list of inputs that are likely to be needed for the subsequent development process is compiled.

Definition of the target benefit

The initial step in the development process is to create a collection of potential roles in which employees could encounter the system to be developed. Generally, and especially in the context of ProKInect, AI functionality is aimed at optimizing existing applications. Therefore, the initial approach for defining roles can start at the current portfolio of roles involved in the application. If applicable, suitable roles can also be sourced from the standards associated with the application. If a large number of roles have been identified, it may be useful to cluster them and agree on the delimitation of the roles within the team. In ProKInect, such roles were collected for the application area "operation of a production plant."

The following roles can be found at the operating company, as well as at the machine manufacturer and the component supplier:

- member of the maintenance or service team
- expert for condition monitoring
- maintenance or service manager
- member of the engineering team

The following roles are only found at the operating company:

- operator
- operating engineer
- production planner
- operations management

This collection is conducted in a brainstorming session on the online whiteboard with the objective of including as many individuals as possible, beyond the development team.

A list of ideas is compiled based on the roles and their associated benefits for the development of the system. The brainstorming process is based on a method for requirements collection. Participants are asked to adopt the perspective of various roles involved with the target system and describe the tasks they would like the system to fulfill, with the goal in mind. These ideas are collected directly on the online whiteboard in the form of a statement from the perspective of the role, such as "As a [role], I would like [task] so that [goal]." Subsequently, the benefit descriptions are clustered and, if necessary, merged during follow-up meetings and their preparation. This process yields a collection of benefit descriptions with a similar level of detail. The collection brings together similar ideas without excluding any of the ideas found. A ranking is then created from this collection, with input from other groups involved in the project or groups with management functions as appropriate. The target benefit for which the initial functionality is to be developed is then selected. A straightforward coordination within the development team can be employed to create the ranking, although it may be necessary to consider other constraints. In this case, more sophisticated tools are required to create the ranking, such as those based on a pairwise comparison.

It is advisable to further define the benefits, for instance, by relating the target benefits, which are currently described in a general manner, to specific properties, components, or potential errors of the target system. Once this process is complete, a benefit description in the form of a concise prose text will be created, which all parties involved in the subsequent process must agree upon.

In the ProKInect project, for instance, the target benefit had to be demonstrably achievable on the available laser cutting machine. This meant that the fault condition had to be relevant in regular operation, affect the components for which the project partners were responsible, be able to be caused intentionally, and be detectable using sensor technology available via the partners' components. In consideration of the boundary conditions, the collection yielded the following target benefit:

"As part of the service team or the maintenance team, I would like to receive information regarding a backlash that has occurred and problems with the leveling (affected axle and recommended repair measure (adjustment, pinion replacement, gear rack replacement, gearbox replacement, in-depth inspection on site)) at the start of the preparation of my assignment so that I can prepare the corresponding assignment."

Collection of requirements

The next step is to define the requirements regarding the benefits in focus. In this instance, we will employ the CTQ tree method. We have identified the factors and requirements that influence the target benefits. Influencing factors represent elements that contribute to the quality of the benefit for users. Subsequently, one or more requirements are defined for these influencing factors, with the objective of ensuring that the fulfillment of these requirements can be assessed in an objective manner. Figure 1 provides an example of a CTQ tree.

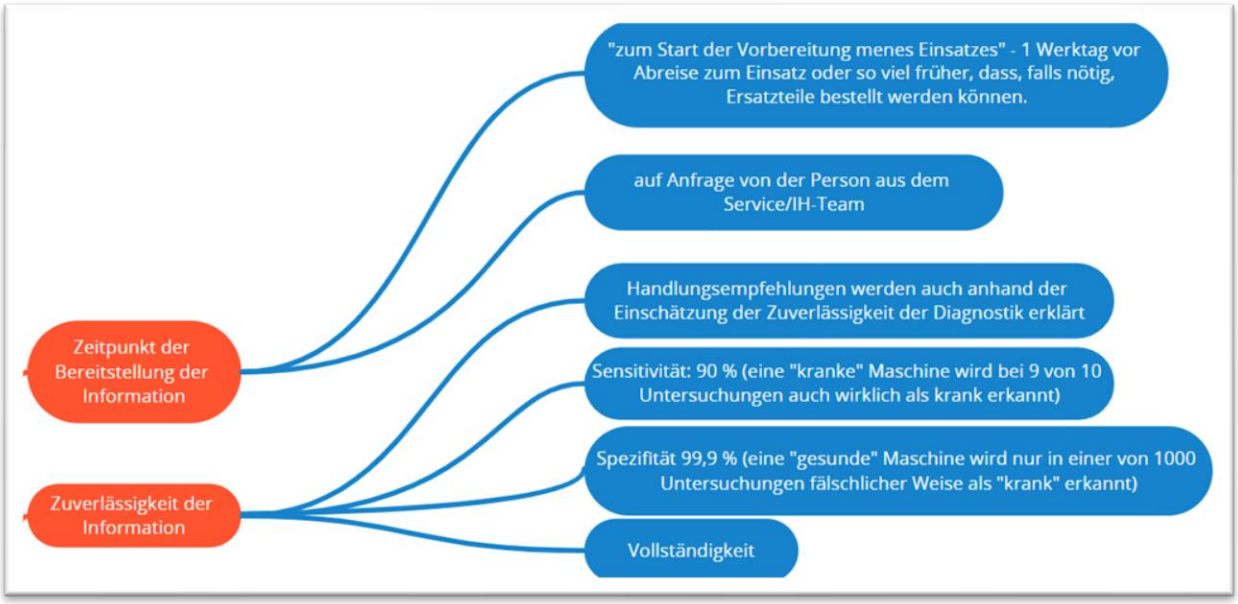

Figure 1: Section of the CTQ tree on the cooperatively created online board; two of the defined influencing factors are shown in red, the associated requirements in blue.

System design

Once the requirements have been collected, the functional decomposition of the defined benefits begins. This process is conducted in parallel with the creation and maintenance of a digitally expandable image of the functional architecture developed in the process. The most effective method for this stage is to continue the cooperative collaboration in online meetings with a shared view of an online board. The software and hardware functions required to provide the defined benefits are broken down from a high-level overview to a detailed, multi-level hierarchy. The starting point is aligned with the data flow, i.e., where the required data originates in physical processes on the real machine. The initial functions in the architecture are those that tap into the physical processes on the real machine. The functions required to provide the desired benefit are then

developed further in the direction of the signal flow. It is essential that the functions collected at the lowest hierarchy level are sufficiently focused to enable the development of at least one, and ideally several, technical solutions. However, the description of the function should not anticipate this solution. The functional architecture is discussed and developed in a collaborative manner, with all parties involved subjecting it to rigorous scrutiny throughout the development process. It is essential to reach a consensus on how the functions interact with each other and how the interaction of the functions can be optimized.

It is to be expected that the development team will first agree on a consensus at the points in the architecture where physical processes are acquired from the machine. Based on these elements, experience has shown that the agreement then continues in the direction of the signal flow. The next phase is then initiated in parallel, commencing with the functions on which there is consensus. This entails determining which partner or individual will be responsible for the subsequent design and, ultimately, the implementation of the selected function. The respective individuals are now responsible for collecting and selecting technical solutions for the function and allocating the technical instances for implementation. It is also important to facilitate an exchange within the project team to obtain a variety of possible solutions.

It is crucial to maintain consensus among the development teams throughout the detailed implementation phase. This is to ensure that the final elements of the functional architecture align with the desired benefits.

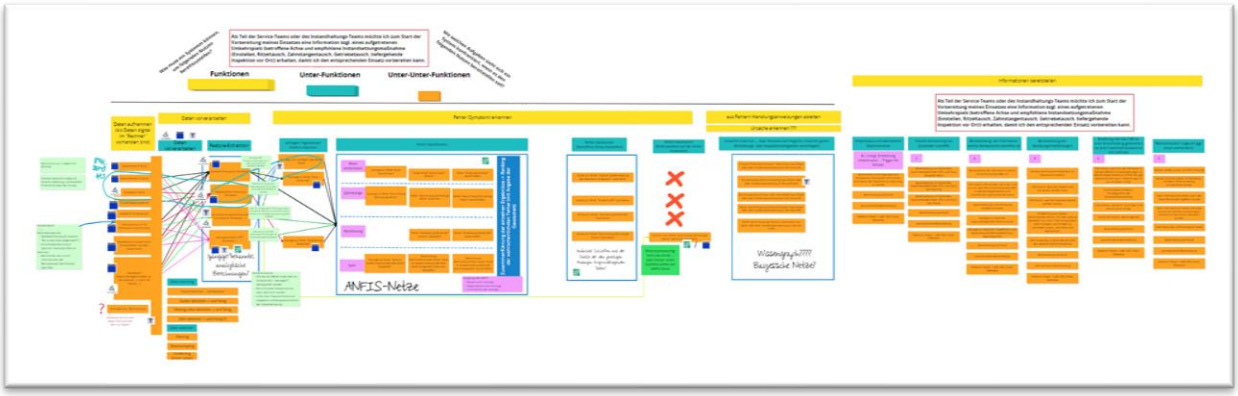

Figure 2: Overview of the section of the online whiteboard with the entire functional architecture

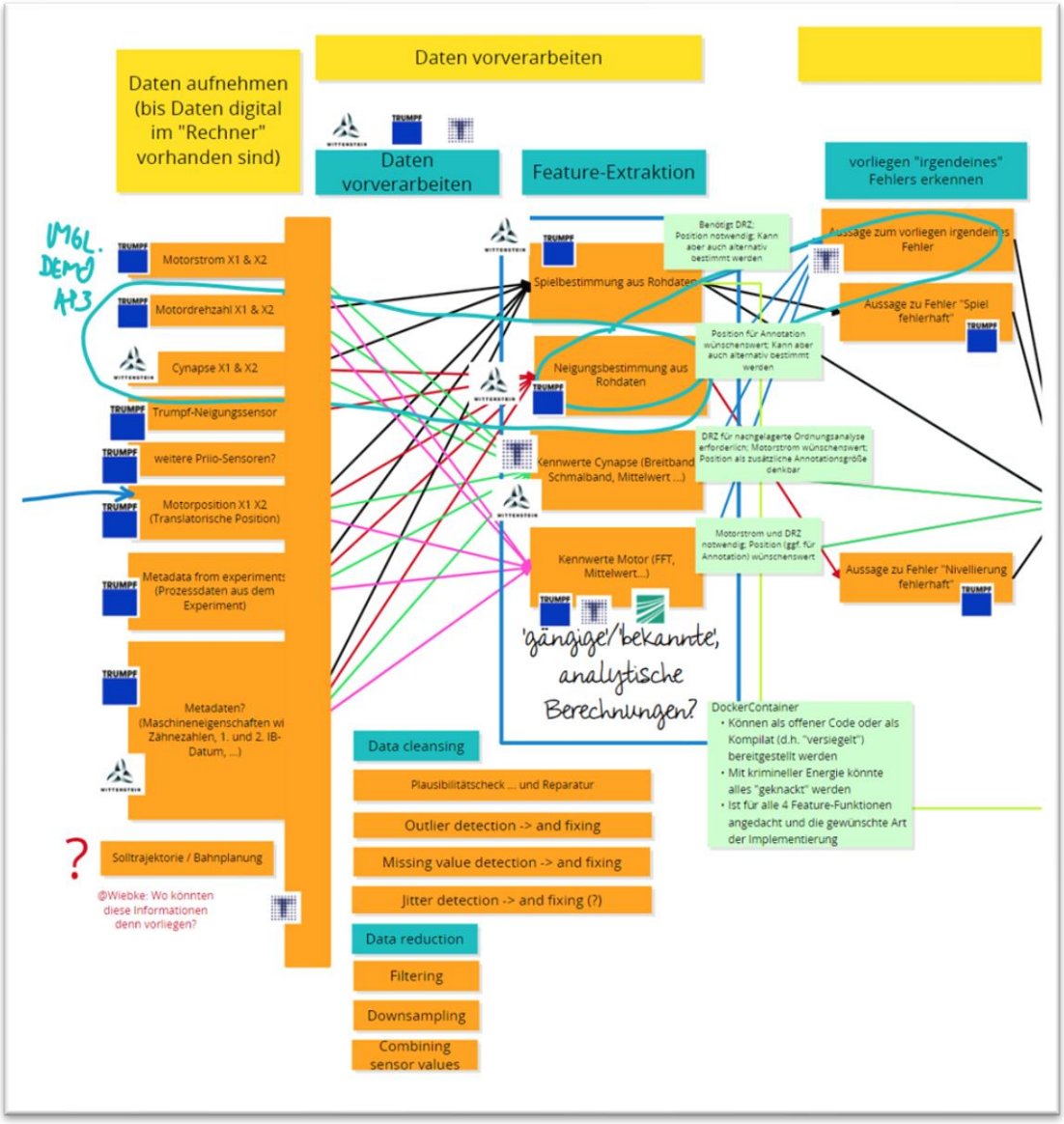

Figure 3: Section of the functional architecture on the online whiteboard

Interface coordination

It is essential that all interfaces are coordinated and documented in a systematic manner. It is the responsibility of the person managing the development team to ensure that this is given particular attention. Along the data flow, there will be interfaces between functions that are developed by different individuals, potentially by different partners, or that are allocated to different technical instances. Extensive and detailed work is required to ensure that the transitions can be technically resolved and coordinated in such a way that the functions can fulfill their respective purposes. The SIPOC representation has proven to be an effective tool for this purpose. SIPOC stands for Supplier, Input, Process, Output, and Customer. It is essential to define these five elements for each individual function. It is the responsibility of the function owner to describe the process and identify the necessary input variables and potential output variables. The supplier and customer are determined by the previously defined links between the functions. In subsequent steps, the respective function owners will consider linked functions and coordinate adjustments to output and input variables at the interface to ensure consistency.

At this stage of the development process, there is a transition in the documentation of the functional architecture. The structure developed on the online whiteboard is transferred into a function graph by the moderator. This offers the option of displaying the interface definition in the SIPOC representation as details of the function blocks. This combines the information from both levels of detail (see Figures 4 and 5).

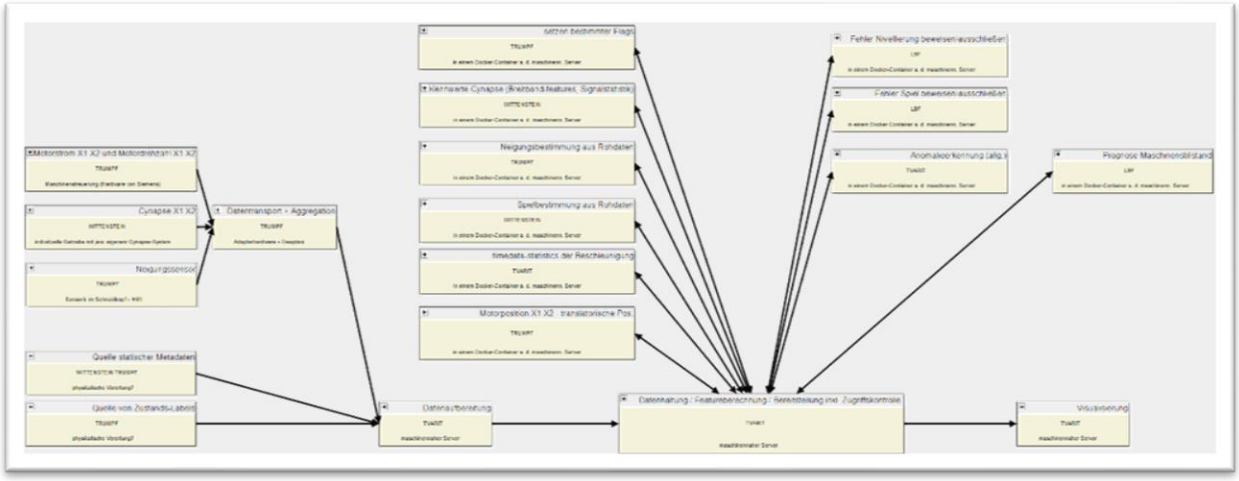

Figure 4: Complete function graph

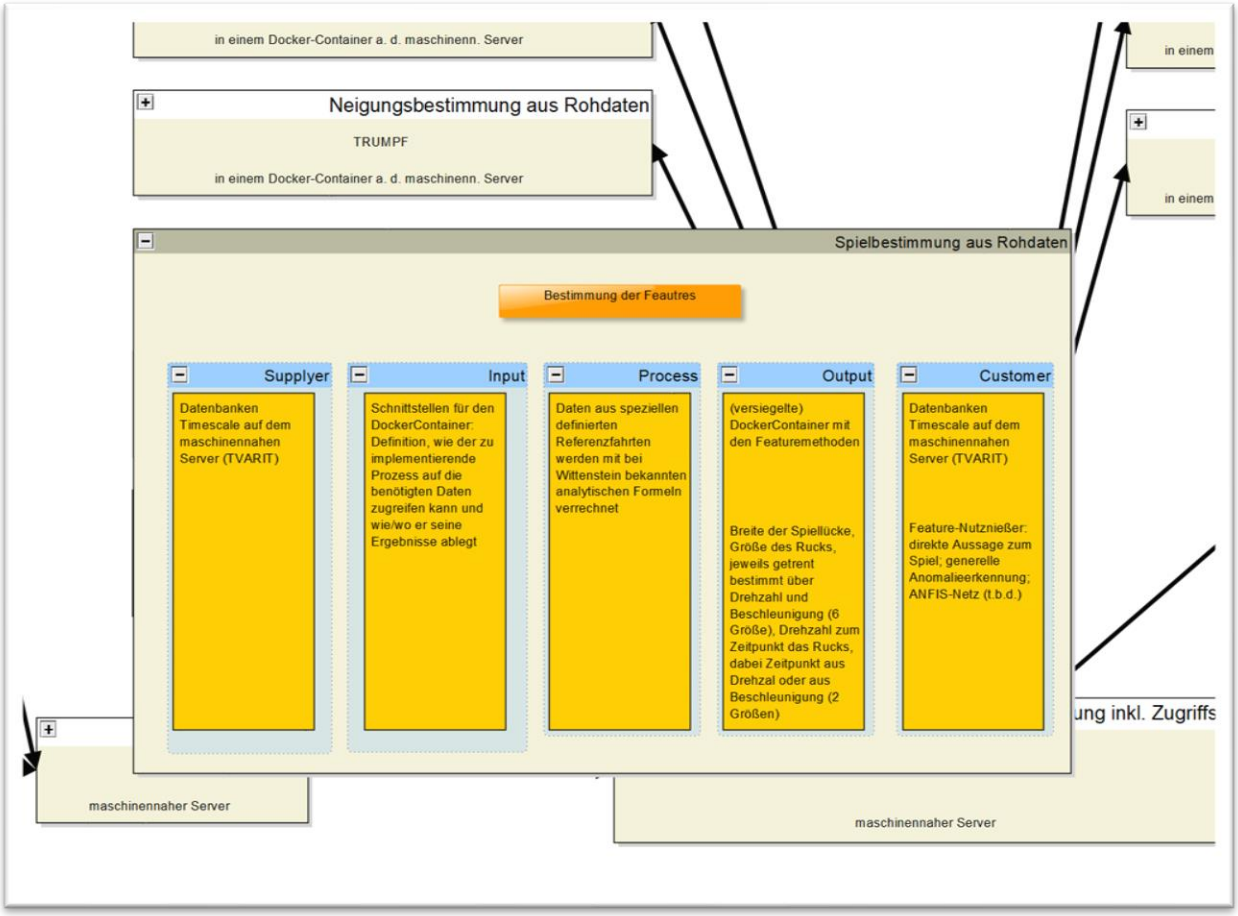

Figure 5: Excerpt from the function graph with SIPOC content

Implementation and support- including clarification of newly arising detailed questions- incl. training phase

The actual implementation of the functions begins step by step with the completion of the individual interface agreements. It is also advisable to initially create the functions as so-called "dummies" on the common architecture. These dummies are, in a sense, empty shells of the functions, but they already read in the agreed inputs and write data structures in the form of the agreed outputs (even if these outputs have not yet been correctly calculated or determined). The function owners are responsible for implementing the dummies and then populating them with the intended functionality. The status of the individual implementation work is reviewed in status meetings. Any challenges arising at the interfaces or during the further processing of input data are also addressed. Where the solutions to these challenges necessitate adjustments to the interfaces, processes, outputs, or even the functional architecture, these are documented in the function graph by the meeting moderator.

Evaluation of the development result based on the collected requirements

Once the development is complete, the results are tested in one or more demonstration cases. These are based on the target benefit and its requirements, ensuring that all requirements can be tested and that the benefit can be demonstrated.

Discussion, outlook and conclusion

In conclusion, the development process for collaborative condition monitoring described here is not significantly different from an agile development and project management process. However, our experience with ProKInect has shown us that there are a few special considerations to be made when implementing such a development process.

It is essential that the development team is unified, has a clear understanding of the intended target benefit, and is focused on it.

While the individual development work steps are relatively short, the effort for coordination and communication is high. This places a correspondingly high demand on the team skills of those involved. It is important to note that moderation should be able to operate independently of content-related work. The effort required to repeatedly scrutinize the details at the interfaces is significant, but it must be undertaken in any case.

Funding: This research and development project is funded by the German Federal Ministry of Education and Research (BMBF) within the “The Future of Value Creation – Research on Production, Services and Work” program (project ProKInect - funding number 02P20A090) and managed by the Project Management Agency Karlsruhe (PTKA). The author is responsible for the content of this publication.