Managing Knowledge of Intelligent Systems
The Construction of a Chatbot using Domain-Specific Knowledge

Marcus Grum(✉), David Kotarski, Maximilian Ambros, Tibebu Biru, Hermann Krallmann, and Norbert Gronau

University of Potsdam, Potsdam 14482, Germany,
mgrum@lswi.de, https://doi...

Abstract. Since more and more business tasks are enabled by Artificial Intelligence (AI)-based techniques, the number of knowledge-intensive tasks increases as trivial tasks can be automated and non-trivial tasks demand human-machine interactions. With this, challenges regarding the management of knowledge workers and machines rise [9]. Furthermore, knowledge workers experience time pressure, which can lead to a decrease in output quality. Artificial Intelligence-based systems (AIS) have the potential to assist human workers in knowledge-intensive work. By providing a domain-specific language, contextual and situational awareness as well as their process embedding can be specified, which enables the management of human and AIS to ease knowledge transfer in a way that process time, cost and quality are improved significantly. This contribution outlines the requirements and a framework to designing these systems and accounts for their implementation.

Keywords: Domain-Specific Language, Morphologic Box, Chatbot

1 Introduction

The capabilities of intelligent systems are increasing with regard to data gathering and analysis. They tap numerous data sources provided in digitized economics contexts, such as business processes and Industry 4.0 environments, and increasingly use AI-based techniques to analyze this data and learn from it [17]. The interplay of humans and machines becomes essential, as all intelligent systems are based on either knowledge representations from AI-based algorithms, by externalized humans experiences or a combination of both [9]. Thus the knowledge transfer between machines and human process participants as well as among process participants is key for an optimal business process and demands for management.

Faced with the task of reducing costs, Ducker even argues knowledge is the only meaningful resource today [4]. Yet, the potential of the adaptability and information processing power of intelligent systems to act as knowledge carrier has not be considered or assessed thus far: If it was possible to design intelligent systems to align with a certain role as process participant in specific knowledge transfer situations and intervene by management-induced interventions, the business processes efficiency can increase, e.g. by means of reduced time or increased quality of process outcomes. For the situational awareness and the selection of interventions, influencing factors need to be known and incorporated into intelligent systems. Further, their behavior needs to be controlled throughout the process. Therefore, the following paper outlines the systematical examination of tools for the creation of AIS, which refer in an example case to a chatbot caring about online customer requests, and the design of a domain-specific language specifying the intelligent system for the optimization of knowledge transfers.

Hence, the following research will focus on optimization of knowledge transfers with help of modeling techniques with the intention to answer the following research question: *"How can intelligent systems be managed with the aid of a domain-specific language?"* This paper intends not to draw an all-embracing description of concrete, technical realizations of those novel management attempts. It intends to set a first step to a clarification of non-transparent knowledge use and its controlling. Hence, sub research questions are:

1. "How can an adequate foundation for the construction of intelligent chatbots be identified systematically?"
2. "How can domain-specific knowledge be integrated with AI-based systems?"

In accordance with a design-science-oriented research [13], the remainder structures as follows. The second section presents a theoretical foundation of knowledge use and managing intelligent systems. The third section identifies requirements for the problem of modeling intelligent systems and a methodological proceeding. These are realized by a design with whom this problem shall be overcome. The fourth section demonstrates the design's functioning, which is evaluated in the fifth section. The final section summarizes the extent to which the initial problem has been solved and the research questions can be answered.

2 Theoretical Foundation and Underlying Concepts

The first sub-section works out an AIS definition, so that an interpretation for modeling chatbots as intelligent systems can be established. Then, knowledge modeling approaches are collected, so that a foundation for the meta-model design is available.

2.1 Chatbots and Artificial Intelligent-Based Systems

Intelligent systems are able to sense their environment, process environmental information such as with the aid of AI-based algorithms, and to respond accordingly by different forms of actuators [7]. The key differentiators of the here calles AI-based Systems (AIS) are the ways how they sense their environment (e.g. through sensors), which kind of powerful AI algorithms are applied (e.g. deep neuronal networks) and how they react to it (e.g. by a certain motion or speech).

Chatbots are a prominent example of AIS these days, which have become increasingly popular for final users as well as businesses. A chatbot is a program that simulates a conversation between a human conversation partner and itself [2]. Here, the chatbot receives an input query from the user (e.g. a question, such as "what is your name?"), it applies different kinds of AI-based algorithms for processing its perception and then responds via its verbalizing actuators by presenting text (e.g. "My name is chatbot." or

"Call me chatbot"). In order to understand and interpret the question as well as being creative to create the response, the chatbot uses some form of individual and external knowledge. Of course, the chatbot considers many more complex components, e.g. for recognizing the conversation partner's intention, translating queries, considering the business' motivation, which are all subsumed under the term of *processing* at the AIS definition presented.

The emergence of chatbots and other intelligent systems powered by machine learning and AI triggered humans to question how such systems think and decide. Out of this, the need for explainability emerged, i.e. explaining the decisions in human terms.

2.2 Modeling Knowledge

To make the most out of the conversations, chatbots need to be able to access knowledge that is distributed within organizations. Typically, it is stored in different silos, such as product catalogues, price lists in Product Information Management (PIM) systems, knowledge about customers in Customer Relationship Management (CRM) systems, and product knowledge or process knowledge in Knowledge Management (KM) systems. So far, the following attempts for modeling knowledge are present.

First, knowledge can be modeled by *propositional logic* or *predicate logic* that evaluate the content of truth of a statement and corresponding predicates [1]. As dialogues surpass the information content of statements to be true and false in everyday life, this knowledge modeling approach is not very attractive for chatbots.

Second, knowledge can be modeled by *rule-based systems* that are able to deal with logical dependencies and relations. These consist of a *factual basis* such as databases, a *rule base* that is probably expert made and provides conditional clauses, as well as an *interpretation engine*, which executes rules on the factual basis [8]. While simple rules-based systems are easy to implement, they lack the level of detail that is necessary to send a high-quality response by AIS, such as the chatbot to be designed.

Third, knowledge can be modeled by *semantic nets* that consider objects, their properties, relations among different kinds of objects and their instances. Here, *frames* extend semantic networks, so that data of that objects are stored by uniform schemes at the corresponding object-specific frame description [15]. Although these networks can be associated with the silos mentioned above, these lack in a human readable clear visualization.

In this sense, *domain-specific languages* (DSL) can be considered as a form of frames referring to high-level software implementation languages that support concepts and abstractions that are related to a particular (application) domain [16]. Since this kind of approach represents organizational knowledge, so that the chatbot can directly process it, the following focuses on the construction of an adequate DSL.

Fourth, *process-oriented knowledge modeling* intends to visually model knowledge, its creation as well as its use along knowledge-intensive processes [6]. Although it has strength in visualization, it lacks in the provision of object information, so that the chatbot can directly process it. Hence, the following focuses on the combination of DSL and process-oriented description languages and to make it accessible for a chatbot.

3 Objectives and Methodology

Following the DSRM approach [13], this section identifies objectives independent from a design. Then, a methodology is presented that satisfies methodological objectives. These are separated from the design and its demonstration, so that artifacts can be created and thereafter, the fulfill of requirements can be evaluated. Following a methodological foundation, designed artifacts give evidence in a demonstration in regard to their functioning.

3.1 Objectives

Aiming to prepare an intelligent chatbot, whose knowledge can be managed with the help of a DSL, this section presents a set of requirements that has to be considered at the artifact realizations.

1. An adequate foundation of tools required for the construction of a chatbot needs to be systematically and methodically identified.
2. Different organizational knowledge sources are to be considered at the chatbot's dialogues. Here, we can find silos of PIM, CRM and KM systems e.g.
3. Knowledge of intelligent systems, that are intended to be managed, needs to be visualized, so that human managers are able to comprehend the effect of their management activities on systems managed.
4. Information and data required for the performing of intelligent systems need to be integrated on a common technical level, so that a chatbot directly can process it.

Based on these requirements, a methodological foundation focuses on a morphological analysis and a design-oriented artifact creation is intended.

3.2 Morphological Analysis

Following Zwicky, in order to explore all the possible solutions to a multi-dimensional, non-quantified complex problem for various domains, the *morphological analysis* is a suitable tool [19, p. 34]. It is accepted in various domains, such as anatomy, geology, botany and biology, and Ritchey summarizes the history of morphological methods [14].

By proceeding the general morphological analysis, the morphologic box, the so called *Zwicky box*, is constructed in five iterative steps [18]: The problem *dimensions* are properly defined first. Since these probably refer to relevant issues, a practical applicability is supported. Then, *parameters* are defined as a spectrum of values for the dimensions. Usually, these refer to different solution approaches for that dimension. Third, by setting the parameters against each other in an n-dimensional matrix, the morphological box is created. Since each cell of the n-dimensional box represents one parameter of the problem, the selection of one parameter per dimension marks a particular state or condition of the problem complex. The selection is called *configuration* and represents one solution of the problem complex. In our case, an empirical survey has been realized to identify the best configuration having the widest acceptance. A fourth step scrutinizes and evaluates possible solutions in regard to the intended purpose. In our case, this is the identification of a best foundation for the chatbot construction. In a fifth step, the optimal solution, which is the morphologic box, is practically applied. If necessary, insights from the application are considered in previous steps.

4 Design

In accordance with the DSRM of Peffers [13], this section presents artifacts that are designed in order to overcome the problem of modeling AIS. Here, the first sub-section presents the design of the systematic selection of tools being relevant for the implementation of an AIS. Then, the draft for a DSL is presented, with whom the technical understanding of AIS is described. The third sub-section designs a human-readable form of visualization the DSL.

4.1 Design of a Morphologic Box

In order to systematically identify the requirements in different dimensions, a morphological box was constructed. The dimensions examined during this process were: Technology, Knowledge, AI, Customers, Vendors, Finance, Politics, Ethics. For each dimension, corresponding properties were compiled and concreted by appropriate guiding questions. Subsequently, different scales per dimensions were recorded and evaluated by means of a consensus to determine whether there are conditions that are mutually exclusive. Finally, the possible scales were examined by means of a survey of experts for the widest acceptance to each. The best configuration of the morphological box can be found at Fig. 1.

4.2 Design of a DSL

In order to enable a dialogue-based system to answer queries in high quality, it must have access to operational knowledge, such as product, customer and process knowledge. This knowledge is available in various sources; Parts of this knowledge are found in product catalogues, price lists and CRM systems as well as in various ontological levels, such as departments, sales and service staff. It is therefore necessary to bring together the extensive operational knowledge by means of knowledge engineering and make it available in the dialog system.

The necessary linguistic knowledge for dialogue design and the situation-specific knowledge should be prepared and made available for a dialogue-based query. For this purpose, a suitable representation should be developed that enables optimal dialogue-based processing. One possibility of representation is a DSL. A DSL is a suitable linguistic knowledge representation that targets a situation-specific problem, instead of general software problem [5]. Then, the dialogue-based system uses the data from the DSL to answer complex questions in dialogue with the customer and can be seamlessly integrated in the operational context.

There are a number of open source tools to design DSLs. Some of these tools include *JetBrains MPS*, *Xtext*, and *TextX*. In this work, the language used for the DSL

Dim. Category	Dimension	Guiding Question	Scale
			Finally Selected
Technical Dimensions	Presence of digital representation of systems	*Does the system provide a digital representation?*	Yes/No
	Live access to systems	*Is the system able to deal with real-time data?*	Real-time data
	Ability of systems to communicate (technique)	*Is the system able to communicate?*	Yes/No
	Ability of systems to communicate (standard)	*Is the system able to communicate?*	Yes/No
	Ability of systems to communicate (interfaces)	*Is the system able to communicate?*	Yes/No
	Ability of systems to sense	*Is the system able to sense its environment?*	Yes/No
	Ability of systems to carry out actions	*Is the system able to interact with its environment?*	Yes/No
	Ability of systems to process	*Is the system able to process data?*	Yes/No
	Requiredness of additional setup	*Do we need any additional setup (e.g. client application, third-party connections, etc.)?*	Yes/No
	Presence of API	*Can we access the data via an API?*	Yes/No
	...	...	...
Knowledge Dimensions	Availability of data	*Is the system able to interact with its environment?*	Yes/No
	Availability of explicit knowledge	*Is the data easily available?*	Yes/No
	Availability of tacit knowledge	*Hard to identify? Structured form?*	Yes/No
	Role as knowledge carrier in process	*Does the system participate in the process, so that it uses its knowledge? Does it have a role in the process?*	Yes/No
	Degree of articulation [German: Artikulationsgrad	*Is it hard to explain the function?*	Numeric Scale of KMDL3.0 (p. 46-50)
	Degree of generality [German: Allgemeinheitsgra	*How general is the knowledge? Can it be used in multiple dialogues?*	Numeric Scale of KMDL3.0 (p. 46-50)
	Competence [German: fachliche Einsicht]	*Does the system address competences for the dialog?*	Yes/No
	Experience	*Does the system address experiences for the dialog?*	Yes/No
	Documentation availability	*Is the product completed by a great documentation?*	Wiki pages
	Sample data availability	*Can we access sample data?*	Yes/No
	...	...	...
AI Dimensions	Input data availability	*Does the system support unsupervised learning?*	Explicitly indicated
	Unsupervised functional approaches	*Does the system support unsupervised learning?*	Yes/No -> Availability
	Input data availability	*Does the system support supervised learning?*	Explicitly/Implicitly/Not indicated at all
	Output data availability	*Does the system support supervised learning?*	Explicitly/Implicitly/Not indicated at all
	Supervised functional approaches	*Does the system support supervised learning?*	Availability / Not Available
	State data availability	*Does the system support reinforced learning?*	Explicitly/Implicitly/Not indicated at all
	Action data availability	*Does the system support reinforced learning?*	Explicitly/Implicitly/Not indicated at all
	Reward data availability	*Does the system support reinforced learning?*	Explicitly/Implicitly/Not indicated at all
	Punishment data availability	*Does the system support reinforced learning?*	Yes/No
	Reinforced functional approaches	*Does the system support reinforced learning?*	Availability/ Not available
	...	...	...
Customer Dimensions	Perceptability in dialog system	*How can data be perceived in the dialog system?*	Obviousness / Questionable
	Usefulness of data	*How might this data be useful in a conversation with the chatbot?*	Yes/No
	SME relevance	*How relevant is this software for SMEs (KMUs)?*	Yes/No
	Expected relevance	*What future relevance for this software do we predict?*	Yes/No
	...	...	...
Vendor Dimensions	System costs	*What is the cost of the system (license, consulting, maintenance, infrastructure)?*	Total costs
	Paid access	*Do we need to pay for (API) access?*	Total costs
	Vendor market share	*What is the market share of the vendor? How many of the clients are SMEs in Germany? How many companies use this kind of software?*	Revenue
	Vendor reputation	*What is the vendor reputation?*	Target group fit
	Vendor business model uniqueness	*How is the vendor's business model unique?*	Long term advantage from software (huge / medium / small)
	Trial version	*Is there a trial developer account?*	Cost
	Technical support	*What is the level of tech support by the vendor?*	Community based
	Vendor technique advantage	*What is the vendor's technological advantage?*	USP in market availability (huge / medium / small / qualitative)
	...	...	...
Financial Dimensions	Presence of data structure	*Does the system provide a data structure, so that the implementation is not expensive?*	Yes/No
	...	...	...
Political Dimensions	Limitations by laws	*Does the use of system-specific data correspond to contemporary laws?*	Yes/No
	...	...	...
Ethical Dimensions	Limitations by labor unions	*Does the use of system-specific data correspond to contemporary ethical understanding?*	Yes/No
	...	...	...

Fig. 1: Morphologic Box with best parameters.

specification is *TextX* [3] per design decision, which is a meta-language model suitable for defining grammar description/rules to build a textual language in Python. Based on the grammar definition, it generates a metal-model and a parser for the language. The expressions of the new language are parsed by the parser and a graph of Python objects which corresponds to the meta-model is built. The main objective is to create and realize the design of a DSL for the specification of complex dynamic linguistic knowledge and complex context-dependent knowledge to be used in a dialog-based system as well as its interactive visualization. Then, a sustainable and methodically secured knowledge acquisition and knowledge utilization can be achieved.

The following words in Table 1 are list of terms that are important for the construction of the DSL grammar. Moreover, the function of each term is explained by the definition provided.

Table 1: List of terms used in DSL grammar definition.

Term	**Definition**
actions	List of things the bot can do or say.
aliases	Alternate name for an entity.
dialogues	Conversation turns between user and bot.
intents	Things we expect the user to say.
products	List of items with properties or other attributes.
responses	Hard-coded values or messages the bot can respond with.
slots	User-defined variables which need to be tracked in a conversation.

As for illustration, Fig. 2 presents the construction of a simple DSL that will serve as a basis for adding more components at a later stage of the development cycle. The grammar definition of each entity or rather component that makes up part of the overall DSL is combined with the remaining ones to generate the full structure.
Each grammar definition is used to interpret and parse the corresponding DSL files.

4.3 Design of a Visualization Concept

Following the most sophisticated knowledge modeling called NMDL [10], the DSL described at section 4.2 is interpreted algorithmically and transcoded to a visualization variant, which corresponds to its subset meta-model of Fig. 3. Since a bidirectional dependency of DSL syntax and the visualization exists, changes of the DSL can be followed up at the visualization and vice versa. By this, it can support the management of AIS [9].

In the meta-model shown, you can see three gray rectangles, each representing an individual perspective on the AIS. The *Process View* characterizes the sequential order of business process tasks, which is realized with the aid of Boolean operators and control flows. Further, it shows which kind of system and which kind of process participant having a certain role is part of that task realization. Thus, it characterizes the behavior of the procedural range of processes and process networks [9].

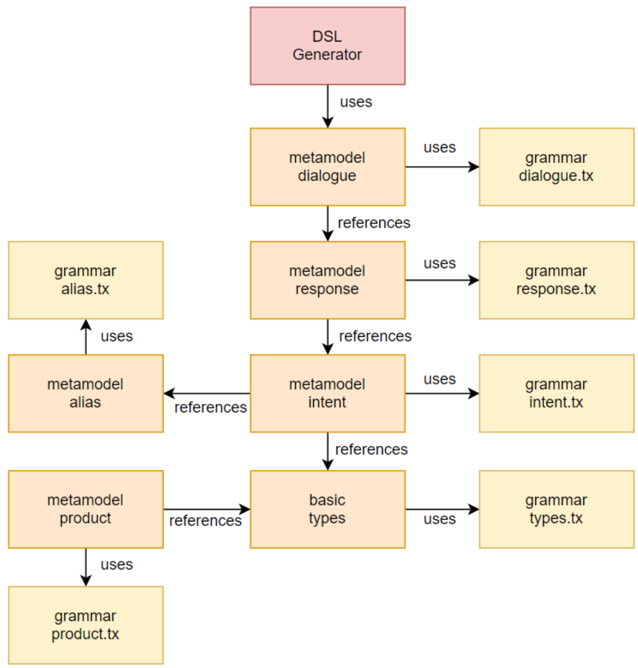

Fig. 2: Generator that combines all defined grammar rules to construct the DSL.

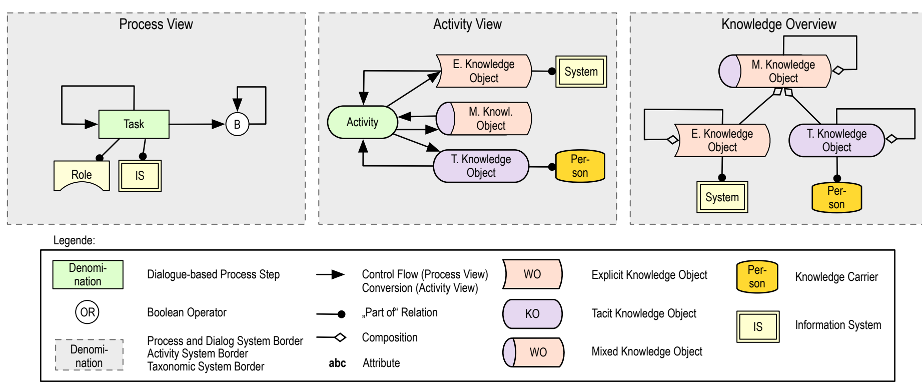

Fig. 3: The DSL meta-model.

The *Activity View* presents the modeling of knowledge transfers among persons and systems. Here, the person-bound forms of tacit knowledge, the person-unbound form of explicit knowledge as well as their combination in mixed knowledge forms is issued, so that knowledge conversions can be clarified in accordance with Nonaka and Takeuchi [12]. The *Knowledge Overview* characterizes the hierarchical decomposition of modeled knowledge objects, so that a comfortable dealing with numerous modeling items is enabled. It also supports the creation of clean and clear models that are easy to interpret.

5 Demonstration

Following the DSRM, this sections applies the artifacts designed to demonstrate its use and evaluate if the initial research problem has been answered. The first sub-section demonstrates the morphologic box in order to identify a best tool foundation. The second sub-section demonstrates the interaction of the DSL and corresponding visualization mechanisms to manage the chatbot knowledge base.

5.1 Identification of an Ideal Tool Foundation

In accordance with the fifth step of the morphological analysis described in section 3.2, the previously established best configuration of the morphologic box has been applied for the identification of a best tool foundation for the chatbot construction. The better a parameter of a tool performs at a certain dimension, the more suitable the tool is for being an adequate foundation of the chatbot construction.

In a workshop session with 12 research and consulting experts of the domains of knowledge management, artificial intelligence, process management and business applications, the guidance questions were answered individually for the software tools that were identified by the workshop participants. In total 22 tools were divided into four clusters, which namely are CRM tools with 5 systems, PIM tools with 9 systems, KM tools with 3 systems and AI tools with 5 systems.

As a consensus of the experts, best tools for each cluster were then determined in the final stage. Preferring the tools that are closest to the previously determined optimum, the detailed analysis can be seen in Fig. 7 at Appendix 3 and the suitability has been visualized by the color range from red (bad) to green (good).

For the chatbot construction, the systems in the CRM, PIM and KM clusters were classified as knowledge sources, so that these can submit required information to the chatbot easily. AI tools rather are used to support the construction of the chatbot. For the CRM cluster, *Hubspot* and *Pipedrive* were chosen because they have a good API and adequate test data availability. In the PIM cluster, *PimCore* was chosen as a priority and extended by *Plytix* because of the vendor business model uniqueness and open source license or rather price model. In the KM cluster, *Confluence* was chosen because of its price model. In the AI cluster, *TensorFlow* and *PyBrian* have been chosen because of the possibility to explicitly consider data at different learning approaches. All these systems thus serve as knowledge base for the chatbot or rather support the basic function of the chatbot.

5.2 Chatbot Use Case

The demonstration of the DSL and its visualization refers to the construction of an AI-based chatbot. Its task refers to the dealing with first contacts of customers on a homepages. Although numerous purposes for customer requests on homepages exist, for the purpose of presenting a clean and clear model, the following focuses on requests about price information of bags.

DSL Example: In order to demonstrate a simple DSL construction based on grammar rules depicted in Fig. 2, an example of a simple conversation exchange between the customer and the chatbot has been prepared by the DSL file definitions at Listing 1.1 up to Listing 1.7. The most illustrative entry point can be found at Listing 1.1, which shows the dialog.

```
dialog:ask_for_product # name:dialog_name
    -> intent:greet
    <- response:utter_greet
    <- response:utter_how_can_i_help
    -> intent:ask_for_bag:
        slot:
            product: bag
    <- response:utter_price_information
        slot:
            price_information: 900
    -> intent:goodbye
```

Listing 1.1: dialogs.dsl example

The different elements used to construct the “dialogs” DSL are as follows:

- “*ask_for_product*” name of dialog guided by intent,
- “→” symbol to identify user utterance,
- “←” symbol to identify bot utterance,
- “*intent:greet*” user utterance with intent ‘greet’
- “*response:utter_greet*” bot utterance with response ‘utter_greet’
- “*response:utter_how_can_i _help*” bot utterance with response ‘utter_help’
- “*intent:ask_for_bag*” user utterance with intent ‘ask_for_bag’
- “*slot: product: bag*” keeps the context of the conversation by storing important piece of information
- “*response:utter˙price˙information*” bot utterance with response of price information
  - “*slot: price_information: 900*” keeps track of context of conversation by storing the customized ‘price_information’
- “*intent:goodbye*” user utterance with intent ‘goodbye’

All the remaining separate DSL file definitions shown in Listing 1.2 up to Listing 1.7 are interwoven with this listing and complement the dialog presented by the provision relevant knowledge from PIM, CRM and KM systems selected (see Section 5.1). Further, they store knowledge acquired in the conversation via slots directly at the DSL, so that the chatbots operational knowledge is accessible in addition.

Visualization Examples: Following the meta-model design presented at Fig. 3, the following presents the modeling of the AI-based chatbot. It is based on the DSL files just issued.

Fig. 4 clarifies the behavior of the chatbot, which is derived from the underlying DSL file of Listing 1.1. Although the following focuses on the DSL visualization, because of the direct dependencies of both, modifications at the visualization are projected on the DSL and vice versa.

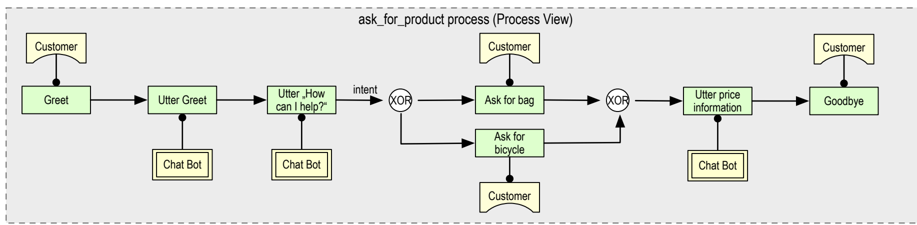

Fig. 4: The process view of the chatbot example.

In the figure, one can see that the conversation with the chatbot is started by the customer (task called *Greet*). Having uttered the greeting by the chatbot (task called *Utter Greet*), it aims to find out the reason for the customer's visit (task called *Utter 'How can I help?'*). In the case the customer asks for a bag (task called *Ask for bag*, the chatbot first presents the corresponding price information (task called *Utter price information*).

The detailed knowledge transfers have been specified by the activity views of Fig. 5, which have been derived from the underlying DSL files as follows:

- *Activity View 1* mainly gets information on the basis of Listing 1.4,
- *Activity View 2* considers information of the Listings 1.3, 1.4 and 1.7,
- *Activity View 3* obtains information of Listing 1.5,
- *Activity View 4* mainly bases on Listing 1.2 and 1.5.

Since the system borders of activity views provide the same naming than the associated task, each activity view specifies the knowledge transfers of a certain task. As the bold written attributes of the four activity views show, the concrete values of process instances are assigned via slots. By this, the concrete process instance of each customer - chatbot - conversation can be visualized. Further, it becomes clear how different systems interact. For instance, facing the activity called *Utter price information*, it becomes clear that three kinds of systems interact in order to deal with the customer's request. The CRM system provides information about the current offer, the PIM system provides the original price information of the bag requested, and the chatbot provides the concrete product request and presents the reply for the customer request. So, the customer gets to know relevant, customized properties regarding the initial request called *desires of bag*, which is with 900 *Euro* less than the standard price of 1.000 *Euro*.

The complete knowledge overviews can be seen at Fig. 6. Since the knowledge objects provide the same naming than the activity views, the concrete knowledge, data and information use at each activity and its corresponding task becomes clear. As the bold written attributes of the three knowledge overviews show, the concrete values of process instances are taken from the activity views. By this, the knowledge base of the chatbot presents the knowledge about each customer. For instance, facing the overview called *chatbot knowledge base*, it becomes clear how a certain customer has been greeted (object called *greeting* having the attribute 'hey') and which information has been presented by the chatbot (object called *product request*).

Let's assume to have the management intend that a certain goodbye phrase shall be removed or used at the dialogue by the chatbot, the corresponding modeling object can simply be deleted or associated with the corresponding modeling object of the knowledge base Fig. 6, which is in this case the object called *goodbye*. Further management attempts refer to the modification of the dialog flow of Fig. 4 and shall be oriented to the symbiotic knowledge management approach of human and artificial knowledge bearers [9].

6 Evaluation

In accordance with the DSRM of Peffers [13], this section evaluates of the demonstration issued in the previous section, if requirements are fulfilled, that have been presented at section 3.1.

- Req. 1 has been satisfied because the methodology of a morphological analysis has been realized in order to create a tool for analyzing different kinds of tools for their suitability to be a foundation of the chatbot creation. By applying the empirically verified morphologic box and identifying a consensus on the evaluation of attractive tools from the viewpoint of practical experts, an adequate foundation has been systematically and methodically identified.
- Req. 2 has been satisfied because the silos *Hubspot* (CRM system), *PimCore* (PIM system) and *Confluence* (KM system) have been considered as organizational knowledge sources and they have been considered at the chatbot dialogue realization using the AI tools called *PyBrain* and *TensorFlow*.
- Req. 3 has been satisfied because the visualization concept and the DSL directly depend on each other. Since the modification of the DSL is directly visualized at the human readable knowledge visualization, and human modifications are directly is transformed to the DSL, the effect of management activities on the chatbot can be comprehended.
- Req. 4 has been satisfied because the different knowledge sources have been projected onto the DSL, which functions as common technical level. So, the chatbot can directly process on the DSL for performing AI-based components realizing the dialogue.

Since requirements have been jointly satisfied, and the demonstration has shown the practical application of the artifacts designed, the modeling and management of AI-based intelligent systems has been shown exemplary at a simple dialogue situation. Next implementation steps will address more complex situations and proof the functioning of sophisticated chatbot versions.

7 Conclusion

The first research question (*"How can an adequate foundation for the construction of intelligent chatbots be identified systematically?"*) can be answered by the conductance of a morphological analysis. On the basis of an empirical research, most relevant parameters of the identification of an adequate foundation for the chatbot construction has been identified. By using the morphologic box constructed, best tools for have been identified, which refer to *Hubspot* (CRM system), *PimCore* (PIM system), *Confluence* (KM system) as well as *PyBrian* and *TensorFlow* (AI system).

The second research question (*"How can domain-specific knowledge be integrated with AI-based systems?"*) can be answered by the concrete DSL design, the visualization concept and the integration of both as follows. Domain-specific knowledge is directly codified at the DSL. This is imported or rather extracted from the knowledge sources, such as *Hubspot*, *PimCore* and *Confluence*, so that it directly can be processed by the chatbot and its AI components. Since the modeling language visualizes the DSL foundation, the chatbot and its AI-based components directly can be manipulated by human readable modeling objects and drag and drop mechanisms.

Although the principle integration has been considered in a practical validation, the technical proof and its validation in everyday situations has not been presented, here.

Next steps of this research will focus on the implementation and the examination of the chatbot in everyday situations. Here, the refinement of modeling objects is evaluated, so that intends, aliases, products, types, responses and dialogue objects can be differentiated. Further, extending the collection of the tool base of the morphological analysis is attractive.

Appendices

1 Domain-Specific Language Listings

```
action:get_product # action:action_name
action:get_reclamation_protocol
action:get_invoice
action:get_offer
```

Listing 1.2: actions.dsl example

```
alias:member # alias:alias_name
    Member
    client
    customer
```

Listing 1.3: aliases.dsl example

```
intent:greet # intent:intent_name
    hey
    hello
    hi slot:first_name # alias:first_name
    good morning
    good evening
    hey there
intent:goodbye
    bye
    goodbye
    see you around
    see you later
```

Listing 1.4: intents.dsl example

```
product:15495 # product:product_identifier
   family: accessories
   category:
    supplier_zaro
    print_accessories
    master_accessories_bags
   name: Bag
   attributes:
     ean: 1234
     weight: 500.00
     price: 1000.00
```

Listing 1.5: products.dsl example

```
response:utter_greet  # response:response_name
    nice to meet you
    hi there
    hi slot:first_name # slot:first_name
response:utter_how_can_i_help
responce:utter_price_information
```

Listing 1.6: responses.dsl example

```
slot:first_name # slot:slot_name
   name: John
```

Listing 1.7: slots.dsl example

2 Visualization of Listings

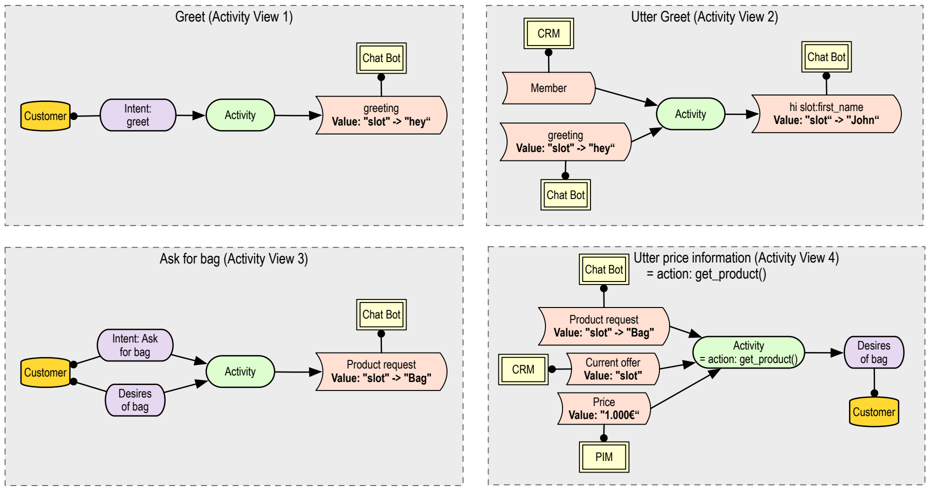

Fig. 5: The activity views of the chatbot example.

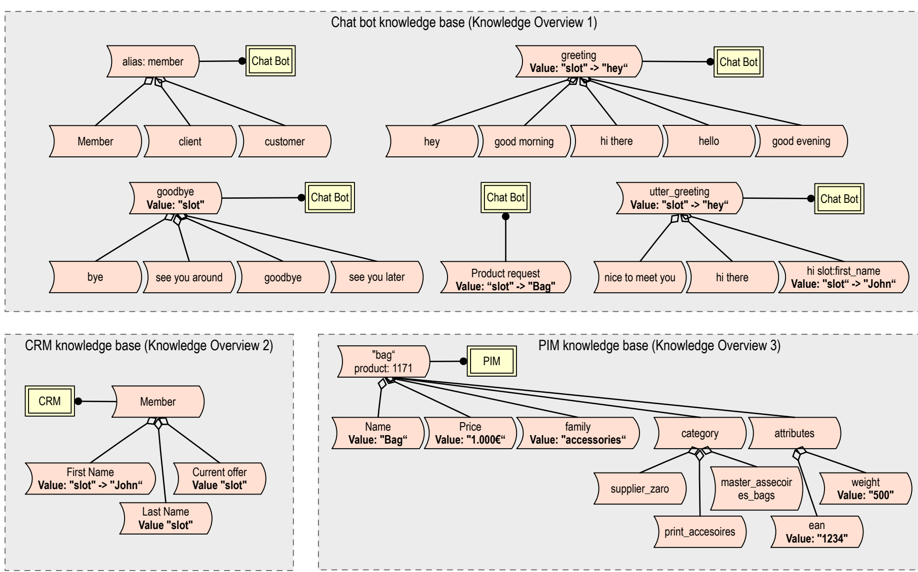

Fig. 6: The knowledge overviews of the chatbot example.

3 Overview of the Tools Analyzed

Dim. Category	Dimension	5) Tool Selection																					
		CRM					PIM									KM			AI Library / Platforms				
		Hubspot	Salesforce	Zoho	Pipedrive	Insightly	Akeneo	Plytix	Salsify	incony	Perfion	sales layer	Pimcore	Productsup	Pimberly	Confluence	Tallium	Comaround	Tensor Flov	PyTorch	Google Dial	Amazon LE	Pybrain
Technical Dimensions	Presence of digital representation of systems	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No	No	Yes	Yes	No
	Live access to systems	Real-time d	Real-time d	Real-time d	Real-time d	Real-time d	Real-time d	Real-time	Real-time	Real-time	Real-time	Real-time	Real-time	Real-time	Real-time	Real-time d	Real-time d	Real-time d	Real-time d	Real-time d	Real-time d	Real-time d	Real-time d
	Ability of systems to communicate (technique)	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No	No	Yes	Yes	No
	Ability of systems to communicate (standard)	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No	No	Yes	Yes	No
	Ability of systems to communicate (interfaces)	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No	No	Yes	Yes	No
	Ability of systems to sense	No	No	No	No	No	No	No	No	No	No	No	No	No	No	No	No	No	Yes	Yes	Yes	Yes	Yes
	Ability of systems to carry out actions	No	No	No	No	No	No	No	No	No	No	No	No	No	No	No	No	No	No	No	No	No	No
	Ability of systems to process	Yes(change	Yes(change	Yes(change	Yes(change	Yes(change	No	No	No	No	No	No	No	No	No	Yes(change	Yes(change	Yes(change	Yes	Yes	Yes	Yes	Yes
	Requiredness of additional setup	No	No	No	No	No	No	No	No	No	No	No	No	No	No	No	No	No	Yes	Yes	Yes	Yes	Yes
	Presence of API	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No	No	Yes	Yes (REST	Yes	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
Knowledge Dimensions	Availability of data	No	No	No	No	No	No	No	No	No	No	No	Yes	Yes	Yes	No	No	No	Yes	Yes	Yes	Yes	Yes
	Availability of explicit knowledge	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No	No	No	No	No
	Availability of tacit knowledge	No	No	No	No	No	No	No	No	No	No	No	No	No	No	No	No	No	No	No	No	No	No
	Role as knowledge carrier in process	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No	No	No	Yes	Yes	Yes	No	No	Yes	Yes	No
	Degree of articulation [German: Artikulationsgrad	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	Degree of generality [German: Allgemeinheitsgra	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0
	Competence [German: fachliche Einsicht]	Yes (salutat	Yes (salutat	Yes (salutat	No	No	No	No	No	No	No	No	No	No	No	Yes	yes	yes	No	No	Yes	Yes	No
	Experience	No (but tags	No (but attri	No (but attri	No	No	No	No	No	No	No	No	No	No	No	Yes (Knowl	Yes (Knowl	Yes (Knowl	No	No	Yes	Yes	No
	Documentation availability	Yes	Yes	Yes	Yes (very g	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
	Sample data availability	Yes	Yes	Yes	Yes (sandb	No	Yes	Yes	No	No	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
	...	...	...	...	...	...	...	...	...				...	...	...	...	...	...	...	...	...	...	...
AI Dimensions	Input data availability	Explicitly	Explicitly	Explicitly	Explicitly	Explicitly	Explicitly	Explicitly	Explicitly	Explicitly	Explicitly	Explicitly	Explicitly	Explicitly	Explicitly	Explicitly	Explicitly	Explicitly	Explicitly	Explicitly	Explicitly	Explicitly	Explicitly
	Unsupervised functional approaches	Not Avail.	Not Avail.	Not Avail.	Not Avail.	Not Avail.	Available	Available	Available	Available	Available	Available	Available	Available	Available	Not Avail.	Not Avail.	Not Avail.	Available	Available	Available	Available	Available
	Input data availability	Explicitly inc	Explicitly inc	Explicitly inc	Explicitly inc	Explicitly inc	Explicitly inc	Explicitly inc	Explicitly inc	Explicitly inc	Explicitly inc	Explicitly inc	Explicitly inc	Explicitly inc	Explicitly inc	Explicitly	Explicitly	Explicitly	Explicitly	Explicitly	Explicitly	Explicitly	Explicitly
	Output data availability	Implicitly inc	Implicitly inc	Implicitly inc	Implicitly inc	Implicitly inc	Explicitly inc	Explicitly inc	Explicitly inc	Explicitly inc	Explicitly inc	Explicitly inc	Explicitly inc	Explicitly inc	Explicitly inc	Implicitly	Implicitly	Implicitly	Explicitly	Explicitly	Explicitly	Explicitly	Explicitly
	Supervised functional approaches	Not Avail.	Not Avail.	Not Avail.	Not Avail.	Not Avail.	Not Avail.	Not Avail.	Not Avail.	Not Avail.	Not Avail.	Not Avail.	Not Avail.	Not Avail.	Not Avail.	Not Avail.	Not Avail.	Available (A	Available	Available	Available	Available	Available
	State data availability	N.i.a.a.	N.i.a.a.	N.i.a.a.	N.i.a.a.	N.i.a.a.	N.i.a.a.	N.i.a.a.	N.i.a.a.	N.i.a.a.	N.i.a.a.	N.i.a.a.	N.i.a.a.	N.i.a.a.	N.i.a.a.	N.i.a.a.	N.i.a.a.	N.i.a.a.	Explicitly	Explicitly	N.i.a.a.	N.i.a.a.	Explicitly
	Action data availability	N.i.a.a.	N.i.a.a.	N.i.a.a.	N.i.a.a.	N.i.a.a.	N.i.a.a.	N.i.a.a.	N.i.a.a.	N.i.a.a.	N.i.a.a.	N.i.a.a.	N.i.a.a.	N.i.a.a.	N.i.a.a.	N.i.a.a.	N.i.a.a.	N.i.a.a.	Explicitly	Explicitly	N.i.a.a.	N.i.a.a.	Explicitly
	Reward data availability	N.i.a.a.	N.i.a.a.	N.i.a.a.	N.i.a.a.	N.i.a.a.	N.i.a.a.	N.i.a.a.	N.i.a.a.	N.i.a.a.	N.i.a.a.	N.i.a.a.	N.i.a.a.	N.i.a.a.	N.i.a.a.	N.i.a.a.	N.i.a.a.	N.i.a.a.	Explicitly	Explicitly	N.i.a.a.	N.i.a.a.	Explicitly
	Punishment data availability	N.i.a.a.	N.i.a.a.	N.i.a.a.	N.i.a.a.	N.i.a.a.	N.i.a.a.	N.i.a.a.	N.i.a.a.	N.i.a.a.	N.i.a.a.	N.i.a.a.	N.i.a.a.	N.i.a.a.	N.i.a.a.	N.i.a.a.	N.i.a.a.	N.i.a.a.	N.i.a.a.	N.i.a.a.	N.i.a.a.	N.i.a.a.	No
	Reinforced functional approaches	Not Avail.	Not Avail.	Not Avail.	Not Avail.	Not Avail.	Not Avail.	Not Avail.	Not Avail.	Not Avail.	Not Avail.	Not Avail.	Not Avail.	Not Avail.	Not Avail.	Not Avail.	Not Avail.	Not Avail.	Available	Available	Not Avail.	Not Avail.	Available
	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
Customer Dimensions	Perceptability in dialog system	Obvious	Obvious	Obvious	Obvious	Obvious	Obvious	Obvious	Obvious	Obvious	Obvious	Obvious	Questionab	Questionab	Questionab	Questionab	Questionab	Questionab	Questionabl	Questionabl	Obvious	Obvious	Questionabl
	Usefulness of data	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No	No	Yes	Yes	No
	SME relevance	Yes	No (tentativ	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No	No	No	Yes	Yes	No
	Expected relevance	No	Yes	Yes	Yes	Yes	No	No	No	No	No	No	Yes	Yes	Yes	Yes	No	No	Yes	No	Yes	Yes	No
	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
Vendor Dimensions	System costs	500€/year/U	150€/year/U	250€/year/U	From 12.50	From $29 p	-	$830/mo	-	-	-	-	Open Sourc	enterprise a	24-72k per	free (up to 1	36$/User/ye	540$/User/Y	0 (No costs	0 (No costs	0 (No costs	$0.00075/re	0 (No costs)
	Paid access	0 (No costs	yes	250€/year/U	0 (No costs	0 (No costs	-	$830/mo	-	-	-	-	0 (No costs	0 (No costs	-	0 (No costs	0 (No costs	660$/User/Y	0 (No costs	0 (No costs	$0.002/requ	$0.00075/re	0 (No costs)
	Vendor market share	-	-	-	1-2%	0.10%	45 Cutomer	-	0.32%	Many	13 in Germa	1 in Germar	<1%	<1%	<1%	<1%	<1%	<1%	0 (Open So	0 (Open So	0.01%	184 compar	0 (Open So
	Vendor reputation	Fit	Fit	Fit	Fit	Fit	Fit	Fit	Fit	Fit	Fit	Fit	Fit	Fit	Fit	Fit	Fit	Fit	Fit	Fit	Fit	Fit	Fit
	Vendor business model uniqueness	Small	Small	Small	Small	Medium (CR	Small	Small	Small	Small	Small	Small	Huge (since	Small	Small	Small	Small	Small	Huge	Small	Medium	Medium	Small
	Trial version	0 (for 30 da	0 (Yes)	0 (Yes)	0 (Yes)	0 (Yes)	0 (Free)	0 (Free)	0 (No)	0 (No)	0 (No)	0 (30 day tri	0 (Open So	0 (No)	0 (No)	0 (free up to	0 (Trial)	0 (Trail)	0 (Open So	0 (Open So	0 (Open So	0 (Open So	0 (Open So
	Technical support	paid vendor	community .	community .	Community	Community	Community	Community	Community	?	?	Community	Community	Community	Community	Community	Community	Community	Community	Community	Community	Community	Community
	Vendor technique advantage	small	medium (a l	small	small	huge (Integ	small	small	small	small	small	small	small	huge (Produ	small	small	small	small	huge (Goog	medium (Ro	medium (Ro	medium (Ro	huge (Proof
	...	...	...	...	...	...	...	...	...				...	...	...				...	...	...	...	...
Financial Dimensions	Presence of data structure	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
Political Dimensions	Limitations by laws	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No	No	No	No	No
	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
Ethical Dimensions	Limitations by labor unions	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No	No	No	No	No
	...	...	...	...	...	...	...	...	...				...	...	...	...	...	...	...	...	...	...	...

Fig. 7: Overview of the analyzed software.

References

1. Ali, A., Khan, M. A.: Selecting predicate logic for knowledge representation by comparative study of knowledge representation schemes, In: 2009 International Conference on Emerging Technologies, Islamabad, Pakistan, 2009, pp. 23-28.
2. Dahiya, M.: A Tool of Conversation: Chatbot. In: International Journal of Computer Sciences and Engineering:158–161 (2017).
3. Dejanović, I., Vaderna, R., Milosavljević, G., Vuković, Z.: TextX: a python tool for Domain-Specific Languages implementation Knowledge-based systems, 115:1-4 (2017).
4. Drucker, P.F.: Post-capitalist Society. Butterworth-Heinemann (1994).
5. Fowler, M.: Domain-specific languages. Pearson Education, (2010).
6. Gronau, N.: Modeling and Analyzing Knowledge Intensive Business Processes with KMDL: Comprehensive Insights Into Theory and Practice. GITOmbh, Berlin, 7 (2012).
7. Gronau, N., Grum, M., Bender, B.: Determining the optimal level of autonomy in cyber-physical production systems. IEEE 14th International Conference on Industrial Informatics (INDIN):1293–1299 (2016).
8. Grosan, C, Abraham, A.: Rule-Based Expert Systems. In: Intelligent Systems: A Modern Approach Springer Berlin Heidelberg:149–185 (2011).
9. Grum, M.: Managing Human and Artificial Knowledge Bearers. International Symposium on Business Modeling and Software Design:182–201. Springer International Publishing (2020). doi: 10.1007/978-3-030-52306-0_12
10. Grum, M.: CoNM Repository. Retrieved 10-2020 from https://github.com/MarcusGrum/CoNM (2020).
11. Hevner, A.R., March, S.T., Park, J., Ram, S.: Design Science in Information Systems Research. Management Informations Systems Quarterly, 28(1):75–105 (2004).
12. Nonaka, I., Takeuchi, H.: The knowledge-creating company: How Japanese companies create the dynamics of nnovation. Oxford University Press (1995).
13. Peffers, K., Tuunanen, T., Gengler, C.E., Rossi, M., Hui, W., Virtanen, V., Bragge, J.: The design science research process: A model for producing and presenting information systems reseach. 1st International Conference on Design Science in Information Systems and Technology (DESRIST), 24(3):83–106 (2006).
14. Ritchey, T.: Problem Structuring using Computer-Aided Morphological Analysis. Journal of the Operational Research Society, Special Issue on Problem Structuring Methods, 57(7):792–801 (2006).
15. Tanwar, P., Prasad, T.V., Aswal, M.S.: Comparative study of three declarative knowledge representation techniques. International Journal on Computer Science and Engineering, 2(07): 2274-2281 (2010).
16. Visser, E.: Generative and Transformational Techniques in Software Engineering (GTTSE 2007), In: In R. Lammel, J. Saraiva, J. Visser (eds.). Lecture Notes in Computer Science. Springer International Publishing, (2008).
17. Waibel, M.W., Steenkamp, L.P., Moloko, N., Oosthuizen, G.A.: Investigating the Effects of Smart Production Systems on Sustainability Elements. Procedia Manufacturing, 8(1):731–737 (2017).
18. Zwicky, F.: Entdecken, Erfinden, Forschen im morphologischen Weltbild. D. Knaur, California (1966).
19. Zwicky, F.: Discovery, Invention, Research - Through the Morphological Approach. The Macmillan Company, Toronto (1969).