A Visionary Way to Novel Process Optimizations
The Marriage of the Process Domain and Deep Neuronal Networks

Marcus Grum(✉) and Norbert Gronau

University of Potsdam, Potsdam 14482, Germany,
mgrum@lswi.de, https://orcid.org/0000-0002-3815-4238

Abstract. Modern process optimization approaches do build on various qualitative and quantitative tools, but are mainly limited to simple relations in different process perspectives like cost, time or stock. In this paper, a new approach is presented which focuses on techniques of the area of Artificial Intelligence to capture complex relations within processes. Hence, a fundamental value increase is intended to be gained. Existing modeling techniques and languages serve as basic concepts and try to realize the junction of apparently contradictory approaches. This paper therefore draws a vision of promising future process optimization techniques and presents an innovative contribution.

Keywords: Process Modeling, Artificial Intelligence, Machine Learning, Deep Neuronal Networks, Knowledge Modeling Description Language, KMDL, Process Simulation, Simulation Process Building, Process Optimization

1 Introduction

A great potential of Artificial Neural Networks (ANN) is well known since nearly four decades. In general, those techniques copy the capabilities and working behavior of the brain in simulating a network of simple nerve cells. Early ANN architectures go back to the 1940s, numerous improvements can be found in late 1980 - 2000 ([34]). Because of their ability to learn non-linear relations, to generalize correctly and to built biologically motivated efficiently working structures, ANN have been applied successfully in various contexts such as music composition, banking issues, medicine, etc. Even simple processes have been modeled on behalf of ANN ([4]).

Nowadays, in times of big data, enormous amounts of data are available and the computing power has increased immensely and with this, the possibility to create bigger and more complex networks. Although, the collection of processing data has become easy, a neuronal modeling and decoding of complex processes has not been realized.

Hence, the following research will focus on deep learning with ANN with the intention to answer the following research question: ”How can the capability to create efficiently working structures of ANN be used for process optimizations?” This paper intends not to draw an all-embracing description of concrete, technical realizations of those novel process optimization techniques. It intends to set a first step to realize the conjunction of the process modeling, simulation and optimization domain on the one hand and the ANN domain on the other hand.

Hence, sub research questions are:

1. ”How can a process modeling language be transported to a neuronal level?”
2. ”How can neuronal processes be modeled?”
3. ”How can neuronal models be used in process simulations?”
4. ”How can neuronal networks be used in order to optimize processes?”

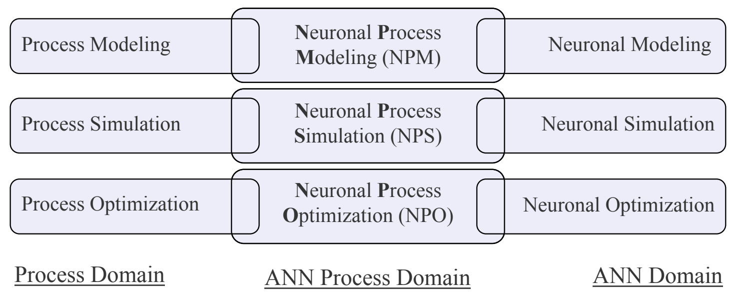


Fig. 1: The ANN process domain as intersection of process and ANN domain.

As Fig. 1 visualizes, the ANN Process Domain is build on the following definitions: A *Neuronal Process Modeling* (NPM) is referred to as the modeling of processes on a neuronal level with a common process modeling language, the reinterpretation of the common process modeling based on that understanding as well as their difference quantity. The *Neuronal Process Simulation* (NPS) is referred to as the process simulation of common process models considering ANN as knowledge model of process participants (persons and machines), the simulation of common process models reinterpreted as deep neuronal network, the simulation of neuronal processes reinterpreted as organizational processes and their difference quantity. The *Neuronal Process Optimization* (NPO) is referred to as common process optimization techniques that are realized on a neuronal level (e.g. double-loop learning on a neuronal level), process optimizations that can be realized because of the learning capabilities of ANN in the domain of common process models as well as their difference quantity.

The research approach is intended to be design-oriented as Peffers proposes ([25] and [26]), such that the paper is structured as follows: A second section presents underlying concepts, the third section derives objectives for a NPM, NPS and NPO. The fourth section provides corresponding designs, followed by their demonstration and evaluation. A final section concludes the paper.

2 Underlying Concepts

Starting with the selection of a modeling approach and the question, how processes can be simulated and optimized in the first subsection, the second section refers to underlying knowledge generation concepts. A further section introduces ANN.

2.1 Process Domain

Following the fundamental procedure model for simulation studies of Gronau ([9]), a model creation is realized after the modeling purpose has been defined, analyzed and corresponding data has been collected. Hence, the following starts with modeling issues. Afterwards, as the model is valid, simulation studies are realized and simulation results collected, analyzed and interpreted. As changes or optimizations are required, adjustments are defined and simulations tested until a sufficient solution has been identified. Best solution options will be implemented, of course. The following characterizes required fields of the process domain following this procedural logic.

Process Modeling. Starting from a basic definition of models, which refer to simpler mappings of relevant attributes of the real world with the intention to reduce the complexity of the real world with respect to modeling objectives, process models can be understood as a homomorphous, time-based mapping of a real-world system focusing a sequence-based, plausible visualization ([9]). According to Krallmann et al. ([21]), a *system* to be modeled consists of an amount of *system elements*, that are connected with an amount of *system relations*. As it is limited by a *system border*, the *system environment* and the system are connected with an interface to exchange *system input* and *system output*.

For the modeling of process systems, several process modeling languages can be used. Considering organizational, behavior-oriented, informational and knowledge-oriented perspectives, Sultanow et al. identify the Knowledge Modeling Description Language (KMDL) to be superior in comparison to twelve common modeling approaches ([37]). Because of the analogy with a human brain as knowledge processing unit, especially a knowledge process modeling is focused. Here, Remus gives an overview of existing modeling methods and a comparison of their ability to represent knowledge ([28, p. 216f]). ARIS, EULE2, FORWISS, INCOME, PROMOTE and WORKWARE are only some representatives. Again, the KMDL can be identified to be superior because of its ability to overcome lacks in visualizations and analyses through the combination of several views such as the process view, activity view and communication view ([14]). This language has been developed over more than ten years. Having collected experiences in numerous projects of various application areas such as software engineering, product development, quality assurance, investment and sales, the evolution of the KMDL can be found in [12]. The current version refers to KMDL version 2.2, but the development of version 3.0 is in progress ([10]). In addition to the modeling language, the KMDL reaches a fully developed research method ([11]).

With its strengths in visualization and the focus of knowledge generation, the KMDL seems attractive for a transfer to the neuronal level. To the best of our knowledge, such a transfer has not been realized yet in any other process modeling language. With its intention to focus on the generation of knowledge following Nonaka and Takeuchi and the intention to transfer the learning potential of ANN, the KMDL enables the modeling of tacit knowledge bases and single or numerous knowledge transfers beside common processing issues. Hence, the KMDL is selected as modeling language for the demonstration in section 5. The current paper builds on the wide spread KMDL version 2.2 ([14]).

Process Simulation. Once, a valid process model has been created, a dynamic sequence and variations of this process can be simulated. Aiming to gain insights within a closed simulation system, the intention is to transfer insights to reality. For this, the following pre-conditions have to be fulfilled: process models have to provide *completeness*. This includes the registration of input data such as time, costs, participants, etc. Further, process models have to provide *interpretability of decisions*. Here, values of variables, state change conditions and transfer probabilities are included. Further, *meta information* have to be considered, as for example the number of process realizations within a simulation. Beneath further objectives, the following *simulation scenarios* can be evaluated quickly and at low costs: current sequences of operations, plans and process alternatives. Those evaluations can be realized before expensive adjustments within current process models (so called as-is models) are implemented in the real world ([9]).

Process Optimization. As processes are adjusted with the intention to optimize them in regard to a certain objective, one speaks from process optimization. All activities and decisions that lead to a desired optimization of business processes, are designated as *business process optimization* ([9]). The success of an optimization is measured by key performance indicators, such as production time, failure and success rates, produced components, etc. There can be found two basic approaches for business process optimizations that are reflected in various methods and variations:

(1) An approach called *Continuous Improvement Process* and

(2) an approach called *Business Process Reengineering*.

(1) Originally inspired by a Japanese living and working philosophy called *Kaizen*, the management concept realizing a never ending improvement of processes and products in small steps is referred to as *Continuous Improvement Process*, (CIP). Following Imai, key principles are a *feedback* mentality and processes are reflected continuously. The everlasting search for *efficiency* demands for the identification and improvement of suboptimal processes, such that waste is reduced and eliminated. Further, the emphasis lies on continual steps rather than giant changes, which is connected to the key principle of *evolution* ([18]). Corresponding management concepts follow cyclic procedures and can be found in numerous variations: Shewhart Cycle ([36]), Deming Wheel ([5]), a second Shewhart Cycle ([6]), PDSA Cycle ([7]), PDCA Cycle ([23]) and a second PDCA Cycle ([19]). In general, those concepts contain planning activities (*plan*). Afterwards, the process is implemented and carried out (*do*). A feedback is collected and compared to a planned output (*check*). Then changes are implemented constantly or revised, before this cyclic procedure is started again. The feedback collection can gather process data either coming from a process realization of the real world, or from a simulated processes. Since improvement ideas are generated during the process realization and a focus lays on single processes, improvements are carried out bottom-up.

(2) The concept of *Business Process Reengineering*, (BPR), refers to the fundamental over-thinking of as-is processes ([15]). This is mostly connected with far reaching changes up to a completely new design of processes and the organization itself. Process improvements are designed as if the organization was built anew and current knowledge and state-of-the-art techniques are considered additionally. Here, improvements are carried out top-down and optimization results provide the following characteristics: Deci-

sions are *decentralized*, process step sequences are *reorganized* and different *process variations* can be considered. Further, the *localization* of working content is organized meaningfully, the need for *control* and required *coordination efforts* are reduced and centralized contact points (e.g. for customer requests) are established.

In conclusion, within the field of process modeling, the modeling language KMDL is suitable for the use as NPM. Since the KMDL does not provide simulation structures, yet, ANN simulation capabilities can be used for an enhancement of the KMDL and enable NPS. Further, learning capabilities of ANN shall be used for the creation as NPM purposes and optimization as NPO purposes. Here, a CIP is attractive for first steps and a BPR can realize further potentials.

2.2 Knowledge Representation

Nonaka and Takeuchi distinguish between explicit knowledge and tacit knowledge ([24]). While the first can be verbalized and externalized easily, the second is hard to detect. The following four knowledge conversion types can be distinguished:

- An *internalization* is the process of integration of explicit knowledge in tacit knowledge. Here, experiences and aptitudes are integrated in existing mental models.
- A *socialization* is the process of experience exchange. Here, new tacit knowledge such as common mental models or technical ability are created.
- An *externalization* is the process of articulation of tacit knowledge in explicit concepts. Here, metaphors, analogies or models can serve to verbalize tacit knowledge.
- A *combination* is the process of connection of available explicit knowledge, such that a new explicit knowledge is created. Here, a reorganization, reconfiguration or restructuring can result in new explicit knowledge.

With the intention to focus on the potentials of human brains and its generation of knowledge, the knowledge generation concepts of Nonaka and Takeuchi seem attractive for modeling on a neuronal level. Since the KMDL is the only modeling language, which builds on this concept, the KMDL is selected for demonstration purposes.

2.3 Artificial Neuronal Networks

Originally, neural networks were designed as mathematical models to copy the functionality of biological brains. First researches were done by Rosenblatt ([32]), Rumelhart et al. ([33]) and McCulloch and Pitts ([22]). As the brain connects several nerve cells, so called *neurons*, by synapses, those mathematical networks are composed of several nodes, which are related by weighted connections. As the real brain sends electrical activity typically as a series of sharp spikes, the mathematical activation of a node represents the average firing rate of these spikes.

As human brains show very complex structures and are confronted with different types of learning tasks (unsupervised, supervised and reinforcement learning), various kinds of networking structures have established, which all have advantages for a certain learning task. There are for example Perceptrons ([31]), Hopfield Nets ([17]), Multilayer Perceptrons ([33], [38], [1]), Radial Basis Function Networks ([2]) and Kohonen maps ([20]). Networks containing cyclic connections are called *feedbackward* or *recurrent networks*.

The following focuses on Multilayer Perceptrons and recurrent networks being confronted with supervised learning tasks. Here, input and output values are given and learning can be carried out in minimizing a differentiable error function by adjusting the ANN's weighted connections. For this, numerous gradient descent methods can be used, such as backpropagation ([27], [1]), PROP ([29]), quickprop ([8]), conjugate gradients ([16], [35]), L-BFGS ([3]), RTRL ([30]) and BPTT ([39]). As the weight adjustment can be interpreted as a small step in a direction of optimization, the fix step size can be varied to reduce great errors quickly. The learning rate decay can be used to reduce small errors efficiently and a momentum can be introduced to avoid local optima. In this stepwise optimization, analogies to continuous process optimizations can be found (see section 4.3).

Since neuronal networks model human brains and the knowledge of a certain learning task, the following refers to neuronal networks as *neuronal knowledge models*. Those represent a current state of knowledge, the capability to generate new knowledge through their activation and interaction and the possibility to transfer knowledge or further process relevant objects within process simulations.

3 Objectives of an ANN Process Domain

As one assumes to have a given process model and one aims to consider a neuronal network as a process participant's knowledge model within the simulation of that process model, the following objectives have to be considered coming from a modeling side:

1. Neuronal knowledge models have to be integrated within existing process models.
2. The same neuronal knowledge models have to be able to be integrated several times within a process model.
3. Neuronal knowledge models have to be integrated within process simulations.
4. Modeled environmental factors (material such as non-material objects) have to be integrated with considered knowledge models.
5. Outcomes (materialized such as non-materialized) of considered knowledge models have to be considered within the process model.

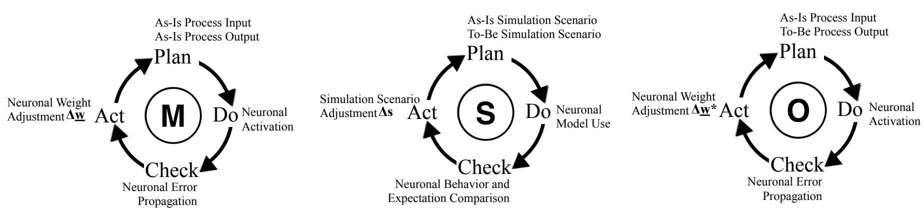
Further, objectives have to be considered coming from a neuronal techniques side:

6. Neuronal tasks have to be considered while neurons follow biological models. This includes both the neuron's everyday business and learning processes.
7. Parallel neuronal task realizations have to be considered within neuronal networks.
8. Time-dependent neuronal behaviors have to be considered within neuronal networks.
9. Sequential neuronal task realization have to be considered within neuronal networks.
10. Different levels of neuronal task abstractions have to be considered in the neuronal process modeling and simulation.
11. Sensory information and knowledge flows have to be considered within the modeled neuronal network.
12. Actuator information and knowledge have to be considered as outcomes of neuronal networks.

Each identified objective of those domains is relevant for the transfer of a process modeling language and serves as input for the following sections.

4 Design of an ANN Process Domain

The visionary way to a novel process optimization is drawn with help of the following subsections. First, a design for a neuronal process modeling is given. Then, a neuronal process simulation design follows. Finally, the neuronal process optimization is designed. All designs refer to the *neuronal process circles* of Fig. 2. Explanations can be found in corresponding subsections.



(a) Neuronal Process Modeling Circle

(b) Neuronal Process Simulation Circle

(c) Neuronal Process Optimization Circle

Fig. 2: Neuronal process circles.

4.1 Neuronal Process Modeling

The following gives definitions of the concept of neuronal modeling. For this, basic definitions are given firstly, definitions based on them are given afterwards.

Neuronal knowledge objects are defined to be neuronal patterns, that evolve as current over a certain period of time that causes a specific behavior of consecutive neurons. Those patterns can reach from single time steps to long periods of time.

Neuronal information objects are defined to be neuronal currents, that serve as interface from and to the environment such as incoming sensory information and outgoing actuator information. Here, stored information is included as well.

Considering those objects, a *neuronal conversion* is defined to be the transfer of neuronal input objects to neuronal output objects. In accordance to Nonaka and Takeuchi ([24]), the following neuronal conversion types can be distinguished:

- A *neuronal internalization* is the process of integration of explicit knowledge (neuronal information objects) in tacit knowledge. Here, experiences and aptitudes are integrated in existing mental models.
- A *neuronal socialization* is the process of experience exchange between neurons within a closed ANN. Here, new tacit knowledge such as common mental models or technical abilities are created.
- A *neuronal externalization* is the process of articulation of tacit knowledge (neuronal knowledge objects) in explicit concepts (neuronal information objects). Here, patterns can serve to verbalize tacit knowledge.
- A *neuronal combination* is the process of connection of available explicit knowledge (neuronal information objects), such that a new explicit knowledge is created. Here, a reorganization, reconfiguration or restructuring can result in new explicit knowledge.

Neuronal input objects are defined to be sensory information objects and knowledge objects.

Neuronal output objects are defined to be actuator information objects and knowledge objects.

An *atomic neuronal conversion* is defined to be a neuronal conversion considering only one input object and only one output object.

Complex neuronal conversion are defined to be neuronal conversions considering at least three neuronal objects of one neuron. *Pure* complex neuronal conversions consider only one neuronal conversion type, while *impure* complex neuronal conversion consider several neuronal conversion types such that one is not able to distinguish them.

Abstract neuronal conversion are defined to be neuronal conversions considering neuronal objects of more than one transferring neuron such that one is not able to identify atomic knowledge flows of participating neurons.

In conclusion, those definitions are the basis for the transfer of process modeling languages to the neuronal level. The logic behind this, which refers to the creation of a practicable, neuronal models, is inspired by standard learning procedures (such as [27] and [1] describe). The neuronal process modeling circle of Fig. 2 (a) intends to visualize this. First, training and test data have to be prepared on base of as-is process input and as-is process output data (*plan*). Then, current ANN are activated by available data (*do*). During the *check*, an activation result is compared to as-is process output data, such that neuronal errors can be generated. Weight adjustments are carried out during the *act* phase. Since this process is repeated until a stopping criteria is reached, a cyclic proceeding is established and results in neuronal process models. Since those can be used within neuronal process simulations, this is the base for NPS and NPO. With this design, an artifact for the first two sub research questions was presented.

4.2 Neuronal Process Simulation

The following gives definitions of the concept of neuronal process simulation. For this, basic definitions are given and a simulation framework is drawn.

For the simulation, common views of the KMDL are brought in a strict hierarchy considering a 1-m-relationship from lower granularity views to higher granularity views. This is visualized in Fig. 3 by a *neuronal process pyramid*.

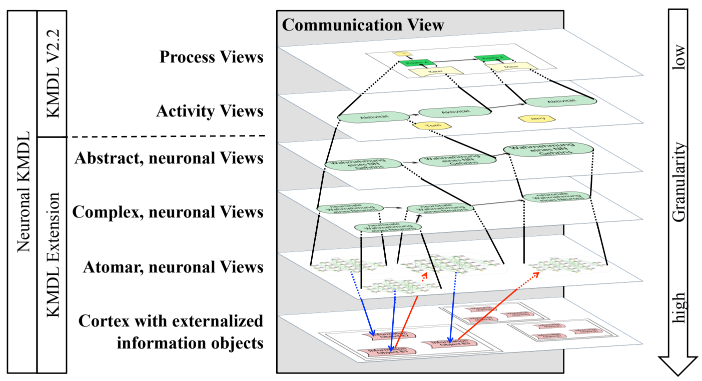


Fig. 3: Neuronal process pyramid.

Here, one can see on the top, that available views of the KMDL (version 2.2) are extended with further neuronal views, as they have been defined in section 4.1. Since all are supplemented with a neuronal simulation logic, they make the *Neuronal KMDL*. The previously mentioned, strict 1-m-hierarchy refers to process views at the very top and knowledge intensive process steps being concretized by a hierarchy of activity views. Although in this figure only one hierarchy is visualized, each process step and corresponding activities are broken down to atomar, neuronal views at the very bottom via various abstract and complex neuronal views. Each represents a collection of connected neurons and are referred to *neuronal subnets* from here on.

A further definition can be found in the realization of a single, discrete *simulation step*. This considers the time-dependent activation of all participating neuronal subnets. Some are activated by former subnets, some by an input of the simulation environment and some by a certain initial input (e.g. for tests by the simulation instance).

The realization of a *simulation sequence* follows the underlying process model of lower granularity levels. Hence, some subnets are not activated at all while some are activated simultaneously and some only under conditions. As some subnets are activated repetitively during the simulation sequence, their current knowledge state can be reused during later activations.

The production of realistic and plausible simulation results requires a *successful training* for participating subnets showing sufficient approximation results and generalization characteristics each. Systematic simulation runs can then be controlled by a *simulation control framework* of a higher order. Here, various simulation scenarios can be realized and compared easily.

The simulation realization is visualized with the neuronal process simulation circle of Fig. 2 (b). It includes the following phases: First, simulation scenarios have to be prepared in order to reflect as-is processes and to-be processes (*plan*). Then, current neuronal models are used on base of corresponding data (*do*). During the *check*, a neuronal behavior is compared to expected results, such that insights can be generated. If more simulations are required, different of optimized neuronal process models need to be used in NPS or scenarios of the next simulation need to be adjusted, then scenario changes are carried out during the *act* phase. Since this process can be repeated until a stopping criteria is reached, a cyclic proceeding is established and results in neuronal process simulations.

In conclusion, those definitions are the basis for knowledge transfers and show knowledge generation and forgetting processes during a neuronal process simulation, which can be compared to a company's intentional behavior. Further, this is the foundation for neuronal process optimizations. With this design, an artifact for the third sub research questions was presented.

4.3 Neuronal Process Optimization

Focusing on CPI corresponding to Kaizen, the optimization of organizational processes and learning processes with ANN have the following in common.

Both require a *set of input factors*. The ANN is activated with a selected set of parameters, which mostly is a codification of real world meanings on base of simulated currents. Organizational processes are fed with input parameters, which are required during the realization of that process.

Further, both produce a *set of output factors*. ANN built on input activations, which are transferred and manipulated in various ways, such that a codification of real world meaning can be generated. Organizational processes combine, manipulate and transform a given input, such that an output is produced. Following the idea of CPI, both kinds of outputs can serve as environmental *feedback* and indicate a fit in planned and achieved performances.

Measured by key performance indicators, the performance of a process is *improved in small steps*. This reflects the CPI idea of *evolution* as follows. Organizational processes are improved by a change of any process parameters: A process can be realized on behalf of better production components, a change in process order, a better qualification of process participants, etc.. This all results in a better process performance. Following Plaut et al. ([27]) and Bishop ([1]), the process of learning with neuronal networks is realized through the adjustment of the network's weights $\triangle\underline{\mathbf{w}}(n)$ in dependence of a prediction error $\underline{\mathbf{E}}$, as Eq. 1 intends to clarify.

$$\triangle\underline{\mathbf{w}}(n) = m \triangle \underline{\mathbf{w}}(n-1) - \alpha \frac{\partial \underline{\mathbf{E}}}{\partial \underline{\mathbf{w}}(n)}, \qquad \text{where } 0 \leq m \leq 1. \tag{1}$$

Here, α is standing for the learning rate, m is standing for the momentum and n is standing for the current training interval. As the performance of the current neuronal model is improved, the error $\underline{\mathbf{E}}$ is reduced stepwise. Here, the following two types of interpretations can be drawn and be connected to CPI-related issues of the *efficiency*:

(1) Interpretations of organizational processes as ANN.

(2) Interpretations of biological processes under economic constraints.

(1) As the improvement of organizational processes is interpreted as a kind of human version of gradient descent method, the desired performance of organizational processes can be interpreted as discrepancy or error $\underline{\mathbf{E}}$, and process changes can follow biological plausible techniques. Hence, an ANN training procedure can be used either to establish required models for neuronal process simulations, improve as-is-process models during the neuronal process optimization or establish new process models in the sense of BPR during the neuronal process optimization. During those optimizations, the following error environments can follow this interpretation plausibly.

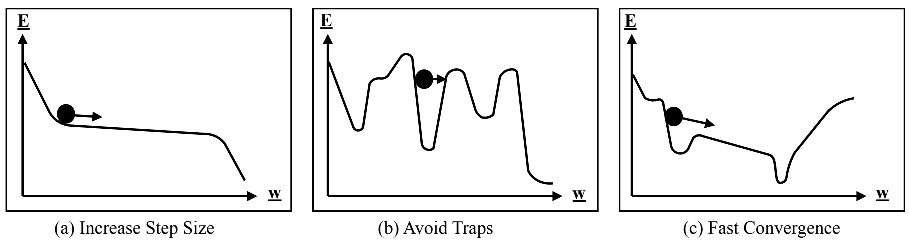


Fig. 4: Error environment examples showing heterogeneous characteristics.

Fig. 4 shows three commonly known error plateaus, which either can stand for the prediction error reduction of ANN during the training process or for discrepancy reduction of a desired performance of organizational processes during the improvement. The current optimization level is visualized by a black ball rolling metaphorically to the global optima. The optimal direction of that moment is indicated by an arrow.

As the learning rate α is adjusted intelligently, numerous cost intensive training runs can be avoided. Fig. 4 (a) shows that the increase of a small step size can speed up the search for the global optima, which can be found in the very right of the diagram. The reduction of continuous process improvement runs can be achieved with help of the expertize and experience of process experts, who consider better changes and a greater number of changes in one run.

As the momentum m is increased intelligently, an oscillation between local error plateaus can be avoided because of the consideration of recent weight adjustments. Fig. 4 (b) shows that the use of a momentum can help to avoid traps during the search for the global optima because several local optima can be overcome because of this moment of inertia. Those can be avoided through the use of standards in the context of continuous process improvement runs. These can help to disregard current trends, irregularities and invalid runs reasonably.

As the learning rate and momentum are adjusted intelligently during the learning process, an effective optimization can be carried out. Fig. 4 (c) shows that a great step size is reasonable at the beginning but has to be decreased at the end, such that a global optima can be identified and a great step size does not oscillate around the global op-

tima. The efficient reduction of continuous process improvement runs can be realized with help of patterns. Here, best practices, routines and proceeding models can help to identify best process options quickly and implement attractive behaviors of changes reasonably.
(2) As the improvement of ANN processes is interpreted as a kind of economic version of organizational mechanisms, the human brain is restricted by the same economic constraints like time, quality and costs. The desired performance of organizational processes can be injected as $\underline{\mathbf{E}}$ in backpropagation approaches and process changes can follow industry specific techniques.

If organizational processes are optimized in regard to *time*, either the throughput time is reduced (e.g. in production processes) or the number of processes is increased, that are realized in a given period of time (e.g. a production date). Analogically, the human brain tries to map a time-based behavior as it is relevant for a corresponding task, e.g. fishing with a spear. Here, a complex sequence of actuator realizations within specified time frames is essential.

Optimizing organizational processes in regard to *quality* refers to the improve of quality factors that are connected to the process outcomes. As example, one can find a better production surface measured by the rate of broken surfaces per month. Analogically, the human brain tries to shrink prediction errors, such that the rate of correct predictions within the corresponding task can be increased and qualitative better results can be realized.

As organizational processes are optimized in regard to *cost*, the cost-intensive use of resources during the process realization is reduced. This can be connected to the use of less and cheaper materials if available, the reduction of required space if possible, or the realization of further tasks in parallel, etc. Since the learning process of the human brain is limited in space (by the size of the human's head), the positioning of neurons representing a certain task and the creation of further connections also is relevant. Additionally, learning processes and the working of the human brain is cost-intensive. Here, proteins and transmitter are required, whose availability is limited, too. Further, they can not be substituted since cheaper materials are not available at all and a cost-efficient working is essential.

In conclusion, the approximation of tasks with neuronal networks can try to realize a trade-off in a maximal number of learned tasks, their approximation accuracy in time and cost-based constraints. As a meaning of each element of the neuronal network can be mapped to an interpretation in the real world, changes in the neuronal network during the learning process can be interpreted directly within the corresponding context of their process representation. Being inspired by these analogies, various tools of both sides, the process domain and ANN domain show promising application possibilities within the neuronal process optimization.

The neuronal process optimization circle of Fig. 2 (c) intends to underline this. The preparation of input data coming from as-is processes and to-be process output data coming from to-be simulations is realized during the *plan* phase. Then, current neuronal models are activated by available data of selected scenarios (*do*). During the *check*, an activation result is compared to to-be process output data, such that neuronal errors can be generated. Weight adjustments are carried out during the *act* phase. Since this process is repeated until a stopping criteria is reached, a cyclic proceeding is established and results in optimized neuronal process models.

With this design, an artifact for the fourth sub research questions was presented.

5 Demonstration of an ANN Process Domain

The following subsections show the realization of the neuronal process modeling on behalf of the *KMDL*. For this, theoretic examples and corresponding process models are given, that visualize basic definitions. Then, practical examples follow. Five examples for neuronal process simulations shall visualize the interpretation of process models as deep neuronal networks and clarify the interplay of simulation output and expectable results. Further, two examples demonstrate the realization of neuronal process optimization.

5.1 Theoretical Example Models

The definitions from section 4 are visualized in the following three theoretical examples: Firstly, atomic knowledge conversions on a neuronal level can be found in Fig. 5.

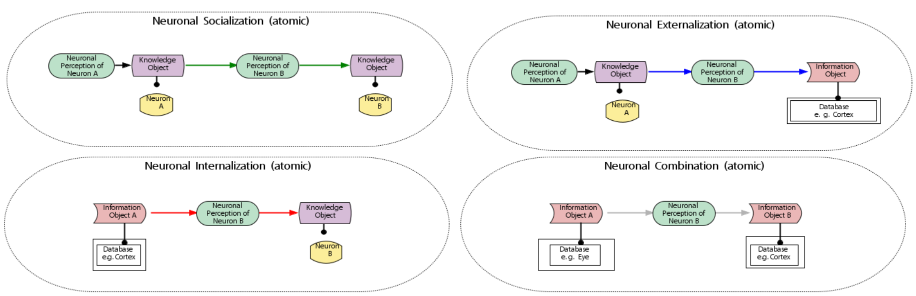


Fig. 5: Atomic neuronal conversions.

In this figure, one can see a neuronal socialization in the top left, a neuronal externalization in the top right, a neuronal combination in the bottom right and a neuronal internalization at the bottom left. All of them were visualized in the activity view of the KMDL.

The entity of *persons* as process participants (yellow) was mapped to *neurons* who interact on a neuronal level. In consequence, the entity of *tacit knowledge objects* (purple) are connected to neurons. The entity of the conversion (green) was mapped to the activity of a neuron that generates new knowledge based on the transfer of its input objects. The environment as well as interaction possibilities with the environment are modeled with the entity of a database (white rectangle). Further, neuronal information objects are stored within a database. In consequence, the shape of *information objects* (red) are connected to those databases.

Secondly, complex neuronal conversions are visualized in Fig. 6.

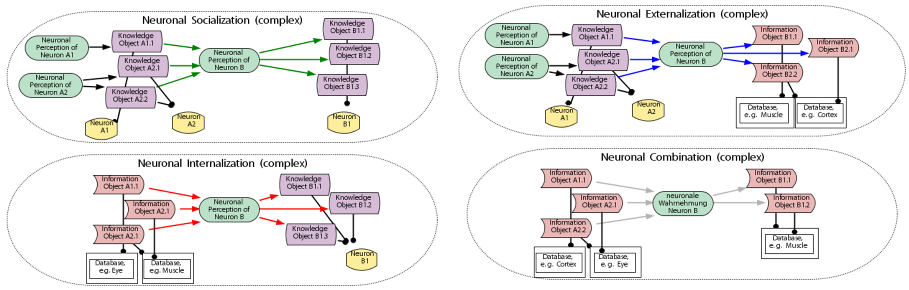


Fig. 6: Complex neuronal conversions (pure).

Again, in this figure, one can see a neuronal socialization in the top left, a neuronal externalization in the top right, a neuronal combination in the bottom right and a neuronal internalization at the bottom left. All of them were visualized in the activity view of the KMDL.

Following the KMDL, conversions of the activity view can be repeated without control flow. Hence, each neuron can develop several neuronal knowledge objects or neuronal information objects over time. Hence, modeled neuronal objects do represent the identified current knowledge of a certain neuron. Therefore, a strict sequence modeling can be realized with help of the listener concept or the process view.

Thirdly, an abstract neuronal conversion can be found in Fig. 7.

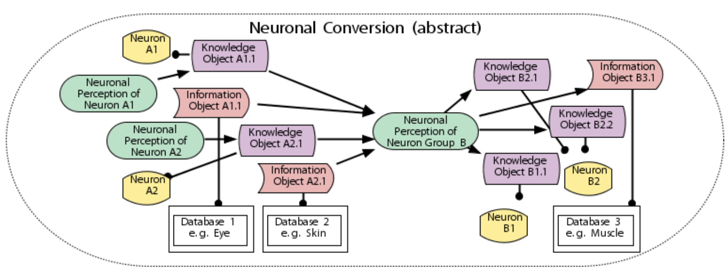


Fig. 7: Abstract neuronal conversion.

In this figure, one can see several impure complex conversions simultaneously, for which reason the visualized arrows are black, as the KMDL asks for. Since more than one neuron (B1 and B2) is considered on that process model, an abstract level of neuronal conversions has been visualized.

With those examples, the first sub research question was answered.

5.2 Practical Example Models

Using basic definitions of a neuronal process modeling, their transfer to practical examples coming from the industry is intended. The following gives four practical examples.

All of them serve as a fruitful domain to visualize neuronal modelings, simulations and optimizations. Their modeling has been carried out on base of the neuronal modeling circle of Fig. 2 (a). Required views look similar to examples of section 5.1, but the labels show concrete meanings.

A first example focuses on the *organization of goods depots*. Those can follow various strategies. For example fix places can hold reservations for certain goods. Alternatively, goods can get an arbitrary place, which considers current free spaces. Here, the human brain can serve as biological inspiration for strategies to store memories and can optimize the depot organization of goods.

A second example focuses on *production processes*. Here, goods are not needed constantly. Meanwhile, they can be stored in goods depots and storage areas. Once, they are needed, they can be brought to the corresponding process step with help of transportation elements ([13]). As they are not required, a transportation element pauses and buffers currently not needed goods. Alternatively, materials can be considered as just-in-time inventory, such that they do not have to be stored in expensive goods depots. Here, the velocity of transportation elements is adjusted in dependence to the production order. Analogies can be found in the human brain. As the storage of goods, the storage of memories can be organized or vice versa. A short-term-memory (current currencies) deals with neuronal knowledge objects similarly to just-in-time inventory. Here, neuronal knowledge objects are used at consecutive neurons as they are needed. Buffered goods are stored within long-term-memories similar to goods depots. Here, currencies are unlocked as they are needed within the current process.

A third example focuses on *specializations of production machines*. As production processes can be considered as a single process network, machines are part of them. Since machines can show high specializations, the organization of production processes can be inspired by the organization of the human brain. Here, certain areas are responsible for a certain task and show high specializations as well. For example the auditory cortex deals mainly with acoustic information, the visual cortex mainly with optical information, etc.

A fourth example focuses on *outsourcing of tasks*: Often, an efficient task realization does not contain the realization of all process steps within the own company. As parts can be outsourced to external parties, analogies can be found in the human brain as well. Here, speed relevant actions can be initiated by reflexes. This is efficient since the realization of a full cognitive task processing would be to slow. As an example, one can imagine the start of a sprinkler system. In case of a fire, it was not effective to create action alternatives, evaluate and select best options but start fighting a fire immediately like a reflex.

With those examples, the second sub research question was answered.

5.3 Practical Example Simulation

On base of the neuronal process simulation circle of Fig. 2 (b), the following example focuses on verifying of the spiral of knowledge of Nonaka and Takeuchi. Their model refers to the broadening of an epistemological knowledge base through the repetitive use of conversions between ontological entities ([24]).

Firstly, the simulation scenario is prepared during the *plan phase*. For this, neuronal process models have been created, as they can be found in Fig. 8.

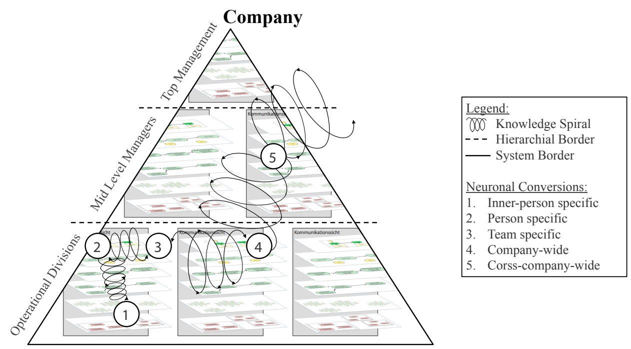


Fig. 8: Neuronal process simulation for spiral of knowledge proof.

Within a triangle representing the system border of an example company, one can find a three-level hierarchy of divisions and corresponding processes by the positioning of several neuronal process pyramids, as they were presented in Fig. 3. Within the example, a first level represents operational divisions, such as sales, production, marketing, etc. A second level stands for processes of mid level managers, being responsible for operational divisions. The top management can be found on the third floor. Within this system border, the simulation scenario realizes the knowledge transfer of an innovation idea that is generated by a person during the production of goods following the company's innovation process. Hence, corresponding simulation parameters are prepared.

Secondly, the simulation is carried out during the *do* phase. Here, five forms of knowledge transfers can be detected. Those were visualized with help of spirals in Fig. 8.

Inner-person specific neuronal conversions can be found on base of the first ontological entity: persons. As a production is realized by the manual work of a single person, an idea of a production process change is generated. Here, a simulation can carry out neuronal conversions of this individual via available views of the corresponding neuronal process pyramid, such that the generation of that idea becomes visible.

Person specific neuronal conversions can be found on base of the first ontological entity as well. Here, the production process innovator follows the underlaying process model and presents the intended process change to a colleague. Their conversation and corresponding conversions can be simulated on base of their individual, inner-person specific process pyramid. Those two are intersecting at the process level. With this, an interaction can be simulated with help of two person specific neuronal networks. The interaction itself can be controlled by an interaction network, characterizing that process step.

Further forms of transfers refer to *team specific* neuronal conversions. Collaborating in teams, the innovator could convince further divisions of his idea in group discussions. Since each participant provides its inner-person specific neuronal pyramid, a group-wise conversion builds on a collective knowledge base as well. This team knowledge is only accessible in this specific cultural circle.

As several teams and divisions are interacting, *company-wide* neuronal conversions can be found at the fourth ontological entity, which refers to the entire company. Following the underlaying process model, a suggestion is prepared for the corresponding mid level manager. This person weights suggestions and prepares contracts for the top level manager. Here, their interaction and the production of corresponding knowledge and information objects can be simulated on base of neuronal process simulations.

Further forms of transfers can be identified at the ontological entity of companies as well, which are called *cross-company-wide* neuronal conversions from here on. As interactions go beyond the company-wide system border, e.g. because of the integration of external consultants in contractual negotiations with the top management and mid level managers concerning cross-company-wide corporations, or open innovation projects, a neuronal simulation can carry out knowledge transfers beyond that border and identify outgoing and incoming objects. With this, an optimal trade-off of chances and risks can be identified.

Thirdly, simulation results are compared with the expected behavior during the *check* phase. Through the simultaneous activation of several neuronal subnets, that are connected by networks representing underlaying current process models, various processes and their interactions can be simulated via deep neuronal network techniques. Those can show surprising side effects, such that a plausibility check is essential. In this example, this refers to the question if the initial innovation idea can be transfered through the repetitive use of conversions between selected ontological entities. When the epistemological knowledge base is broadened through this, the spiral of knowledge can be proven with neuronal process simulations.

Fourthly, if simulation results are not clear and plausible, simulation parameters can be changed during the *act* phase, such that the knowledge spiral can be proved or disproved in next simulation runs.

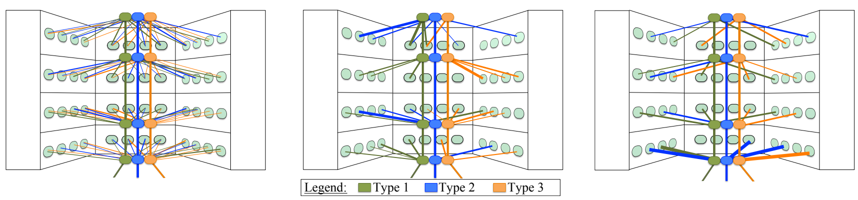
This example shows that the organizational working can be considered as deep neuronal network. Their results of neuronal process simulations serve as reference scenario and build the foundation for neuronal process optimizations. With this example, the third sub research question was answered.

5.4 Practical Example Optimization

The following shows an optimization example, which is limited by given place constraints. This underlines the relevance of a physically interpretable neuron positioning, which is best situated within the real world space. Here, Augmented Reality visualizations become attractive because of their intersection of the real and augmented world. Further, without the possibility to situate models within the real world, 3D models can give a spacial impression of those examples as well.

So, imagine to have a shopping mall with various floors. Those are connected with moving stairs. Here, relevant real world objects can serve for neuronal process optimizations. Hereunder, one can find neurons representing shops. Those are placed inside of the building modeling their real physical position. Further, one can find neurons for moving stairs, which can be found in the building center. Hence, neuronal subnets can stand for available routing points. The movement of customers can be represented by currencies routed from neuron to neuron. Since the shopping mall is limited by walls, the physical size of single shops within the building and the number of routing neurons is strictly limited.

A visualization of the building, the positioning and connecting of routing neurons can be found in Fig. 9. Here, three sub-figures visualize an as-is neuronal process model (NPM) (a) and two neuronal process optimization (NPO) results (b) and (c). Those have been realized following the neuronal process optimization circle of Fig. 2 (c) with different optimization objectives. Neuronal input and output objects have not been visualized in Fig. 9 at all because of a better clarity. Since an optimization shall focus on three types of customers, different colors have been used for each type.



(a) Modeling & Simulation. (b) Optimization 1. (c) Optimization 2.

Fig. 9: Neuronal process optimization in shopping mall.

In Fig. 9 (a), one can see an as-is model, which was created with help of real-world process data and an ANN training process. Here, one can see a positioning of specific shops as it was intended by the shopping mall organizers. The simulation performance of this organization serves as reference point for a neuronal process optimization. Here, the position of real world shops within the shopping mall is questioned. Further, the adjustment of the shop sizes is addressed.

In Fig. 9 (b), one can see the result of a first NPO focusing on the maximization of the shopping mall profit. During the *plan* phase, the training data was manipulated in regard to the desired profit increase. Hence, the corresponding data output represents the intended behavior of to-be processes. Since current neuronal process models did not reflect the desired performance, which became clear because of its activation during the *do* phase, discrepancies were detected during the *check* phase. Then, process optimizations have been carried out during the *act* phase through the adjustment of the network's weights. Those lead to a clearance in the network's connections. Since not all connections were required, a learning and adaption process lead to a deletion of unattractive connections. The different size of connections of the result indicate that some shops are visited more frequently than others and lead to more profit. In general, a cluster for each customer type can be identified. Customer type 1 can mainly be found on the left of the building. Shops on the third floor mainly attract customer type 2 and floor 1 and 2 of the right are attractive for the third customer type. Shop sizes of the real world can be adjusted correspondingly.

In Fig. 9 (c), one can see a second result of a NPO focusing on the attraction of new customers. Here, the training data was manipulated in regard to the desired customer increase during the *plan* phase. Further training runs taking that data into account lead to a different clearance in the network's connections (phases *do*, *check*, *act*). The positioning of the most frequently visited shops (showing the largest connections) can be found on the ground floor and their real world shop size can be optimized. Hence, the shopping mall satisfies most of the customers quickly and attracts new customers efficiently. Further shops that are not attracting a large number of customers have been spread around the entire building, such that bargains can be provoked.

The example describes a neuronal process optimization on base of biologically inspired learning procedures of the human brain. In regard to a specified process improvement dimension, an as-is process was optimized in regard to two directions. A change within a deep neuronal network leads to direct interpretable changes in the real world. With those examples, the fourth sub research question was answered.

6 Evaluation

Faced with the demonstration artifacts of the previous section, objectives of section 3 have been considered as follows.

Objective 1 can be fulfilled by modeling neuronal knowledge models within the activity view characterizing a certain person. Here, a decomposition, such as the neuronal process pyramid it provides, rises the process model granularity of the selected activity and connects all neuronal process models with common process models. Since the common activity view characterizes a corresponding process task of the process view, neuronal knowledge models are integrated within existing process models. Since a neuronal network characterizes entities of persons, a trained neuronal network can be reused in any activity (objective 2). As neuronal knowledge models can be activated and can evolve over time, they can be integrated within discrete process simulations easily (objective 3). From a common activity view modeled environmental factors (material such as non-material objects) serve as interface for the activity view on a neuronal level. Hence, objective 4 and objective 5 are considered as well.

Further, objectives have been considered coming from a neuronal technique side as follows: As learning with neuronal networks is not affected by the here presented concepts, neuronal tasks can follow the neuron's biological models (objective 6) in both, neuronal process simulations as section 5.3 shows in an simulation example and section 5.4 in two optimization examples. A parallel neuronal task realization within neuronal networks has been considered (objective 7) as can be seen in Fig. 6 (neuronal socialization and neuronal externalization) and Fig. 7 (abstract neuronal conversion). Here, at least two neurons realize a parallel task processing. Objective 8 can be met as soon as recurrent connections are considered within the neuronal process models. Then, time-dependent neuronal behaviors are considered within neuronal networks. A sequential neuronal task realization within neuronal networks can be considered within the neuronal process modeling (objective 9), as presented activity views are characterizing corresponding tasks of the process view. Since logical control-flow operators can be used here, a sequential neuronal task processing can be modeled easily. This can be seen in the examples of section 5.3 and 5.4. Further, a time-dependent behavior of a network modeled within the activity view can result in a sequential task processing. Objective 10 has been met as can be seen in Fig. 7. Here, the task "Neuronal Perception of Neuron Group B" models the activity of "Neuron B1" and "Neuron B2" on an abstract level. Further, knowledge objects, information objects, neurons and databases can be grouped and visualized on an abstract level. Sensory information and knowledge flows can be considered within the modeled neuronal network as can be seen in Fig. 5 and Fig. 6. In both figures, possible sensory information flows can be seen at the bottom (neuronal internalization and neuronal combination). Possible knowledge flows can be seen in both figures in the top (neuronal socialization and neuronal externalization). Objective 12 can be met as follows: Actuator information and knowledge have been considered as outcomes of neuronal networks. This can be seen in Fig. 5 and Fig. 6. In both figures, possible actuator information flows can be seen on the right (neuronal externalization and neuronal combination). Possible knowledge flows can be seen in both figures on the left (neuronal socialization and neuronal internalization).

Considering the here presented evaluation of given objectives, it becomes clear that an idea for every objective has been identified. This supports the functioning of the transfer of the KMDL to a neuronal level, such that a neuronal process modeling, a neuronal process simulation and a neuronal process optimization can be built on that base.

7 Conclusion

In this paper, a visionary way to novel process optimization techniques has been drawn and the base has been realized on behalf of the KMDL. Main contributions and scientific novelties are the following: Definitions of a neuronal process modeling, neuronal process simulation and a neuronal process optimization have been created. Objectives of transferring a common process modeling language and a so called ANN process domain have been identified. Further, definitions for those concepts have been created and a modeling language has been transferred to the neuronal world. This includes the reinterpretation of existing shapes of the KMDL. On that base, theoretical examples have been visualized on behalf of the KMDL. Further, analogies for the use of the here presented concepts in the industry context have been identified. With this, the drawn transfer has been applied and proven. Hence, the first sub research question was answered.

The second research question was answered with the design of the neuronal process modeling circle. Its application was demonstrated in the second example, such that industry analogies could have been identified. Further, it was required for the creation of neuronal models, which were used in NPS in the third example. Lastly, it was used for the model creation of the as-is process in the fourth example.

Further, the neuronal process simulation circle and the neuronal process optimization circle were designed and analogies of neuronal learning procedures and process optimization procedures were drawn. Each application was demonstrated in the third and forth example, such that the third and fourth sub research question were answered.

Hence, the following potentials are suitable for next steps: The function concretion of previously presented concepts will be realized. Then, those will be realized as quantitative neuronal process modelings, simulations and optimizations. Further, the comparison of the here presented concepts with traditional results was attractive as well.

The application of the here presented concepts are assumed to cause a radical value increase. As simple and complex relations in different process perspectives like cost, time or stock can be considered, the prediction quality of process simulations can strongly improve going beyond the prediction quality of simple regression models or humans. Further, common optimization potentials can be estimated efficiently. Additionally, new optimization approaches and optimization potentials can be identified.

References

1. Bishop, C.: Neural Networks for Pattern Recognition. Oxford University Press, Inc., ISBN 0198538642, 1 (1995).
2. Broomhead, D, Lowe, D.: Multivariate functional interpolation and adaptive networks. Complex Systems, 2:321–355 (1988).
3. Byrd, R. H., Lu, P.,Nocedal, J., Zhu, C. Y.: A limited memory algorithm for bound constrained optimization. SIAM Journal on Scientific Computing, 16(6):1190–1208 (1995).
4. Chambers, M., Mount-Campbell, C.A.: Process optimization via neural network metamodeling. International Journal of Production economics, 79:93–100 (2000).
5. Deming, W.: Elementary Principles of the Statistical Control of Quality. JUSE (1950).
6. Deming, W.: Out of the Crisis. MIT Press. Cambridge (1986).
7. Deming, W.: The New Economics. MIT Press. Cambridge (1993).
8. Fahlman, S.: Faster learning variations on back-propagation: An empirical study. Proceedings of the 1988 connectionist models summer school, In D. Touretszky, G. Hinton and T. Sejnowski, editors, San Mateo, Morgan Kaufmann, 38–51 (1989).
9. Gronau, N.: Geschäftsprozessmanagement in Wirtschaft und Verwaltung., 2. Gito (2017).
10. Gronau, N., Thiem, C., Ullrich, A.,Vladova, G., Weber, E.: Ein Vorschlag zur Modellierung von Wissen in wissensintensiven Geschäftsprozessen. Technical report, University of Potsdam, Department of Business Informatics, esp. Processes and Systems (2016).
11. Gronau, N.: Process Oriented Management of Knowledge: Methods and Tools for the Employment of Knowledge as a Competitive Factor in Organizations (Wissen prozessorientiert managen: Methode und Werkzeuge für die Nutzung des Wettbewerbsfaktors Wissen in Unternehmen). Oldenbourg Verlag München (2009).
12. Gronau, N.: Modeling and Analyzing knowledge intensive business processes with KMDL - Comprehensive insights into theory and practice. GITO mbH Verlag Berlin (2012).
13. Gronau, N., Grum, M., Bender, B.: Determining the optimal level of autonomy in cyber-physical production systems. Proceedings of the 14th International Conference on Industrial Informatics (INDIN) (2016).
14. Gronau, N., Maasdorp, C.: Modeling of organizational knowledge and information: analyzing knowledge-intensive business processes with KMDL. GITO mbH Verlag Berlin (2016).
15. Hammer, M., Champy, J.: Reengineering the corporation: a manifesto for business revolution. Harper Business. New York (1993).
16. Hestenes, M. R., Stiefel, E.: Methods of conjugate gradients for solving linear systems. Journal of Research of National Bureau of Standards, 49(6):409–436 (1952).
17. Hopfield, J. J.: Neural networks and physical systems with emergent collective computational abilities. PNAS, 79(8):2554–2558 (1982).

18. Imai, M.: Kaizen: The Key to Japan's Competitive Success. McGraw-Hill/Irwin (1986).
19. Ishikawa, K.: What is Total Quality Control? The Japanese Way. Prentice-Hall Inc. (1985).
20. Kohonen, T.: Self-organization and associativ memory. Springer-Verlag New York, Inc., New York, NY, USA, ISBN 0-387-51387-6, 3rd edition:1 (1989).
21. Krallmann, H., Frank, H., Gronau, N.: Systemanalyse im Unternehmen. Oldenbourg Wissenschaftsverlag (2001).
22. McCulloch, W. S., Pitts, W.: A logical calculus of the ideas immanent in nervous activity. MIT Press, Cambridge, MA, USA, ISBN 0-262-01097-6, 15–27 (1988).
23. Moen, R., Norman, C.: Evolution of the PDCA. Google Scholar (2006).
24. Nonaka, I., Takeuchi, H.: The knowledge-creating company: How Japanese companies create the dynamics of innovation. Oxford university press (1995).
25. 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).
26. Peffers, K., Tuunanen, T., Rothenberger, M.A., Chatterjee, S.: A design science research methodology for information systems research. Management Informations Systems, 24(3):45–78 (2007).
27. Plaut, D.C., Nowlan, S.J., Hinton, G.E.: Experiments on learning backpropagation. Technical Report CMU-CS-86-126, Carnegie-Mellon University, Pittsburgh, PA, 1 (1986).
28. Remus, U.: Process-oriented knowledge management. Design and modelling. PhD thesis, University of Regensburg (2002).
29. Riedmiller, M., Braun, H.: A direct adaptive method for faster backpropagation learning: The RPROP algorithm. Proc. of the IEEE Intl. Conf. on Neural Networks, San Francisco, CA, 586–591 (1993).
30. Robinson, A.J., Fallside, F.: The utility driven dynamic error propagation network. Technical Report CUED/F-INFENG/TR.1, Cambridge University Engineering Department, 1 (1987).
31. Rosenblatt, F.: The perceptron: A probabilistic model for information storage and organization in the brain. Psychological Review, 65:386–408 (1958).
32. Rosenblatt, F.: Principles of Neurodynamics. Spartan, New York, 1 (1963).
33. Rumelhart, D.E., Hinton, G.E., Williams, R. J.: Learning internal representations by error propagation. MIT Press, Cabridge, MA, USA, ISBN 0-262-68053-X, 318–362 (1986).
34. Schmidhuber, J.: Deep learning in neural networks: An overview. Neural Networks, 61:85–117 (2015).
35. Shewchuk, J. R.: An introduction to the conjugate gradient method without the agonizing pain. Technical Report, Carnegie Mellon university, Pittsburgh, PA, USA, 1 (1994).
36. Shewhart, W.: Statistical Method from the Viewpoint of Quality Control. Wiliam Edwards Deming. Dover Publications Inc. New York (1939).
37. Sultanow, E., Zhou, X., Gronau, N., Cox, S.: Modeling of processes, systems and knowledge: a multi-dimensional comparison of 13 chosen methods. International Review on Computers and Software (IRECOS), (6):3309–3319 (2012).
38. Werbos, P. J.: Generalization of backpropagation with application to a recurrent gas market model. Neural Networks, 1 (1988).
39. Williams, R. J., Zipser, D.: Gradient-based learning algorithms for recurrent networks and their computational complexity. In Y. Chauvin and D.E. Rumelhart, editors, Lawrence Erlbaum Publishers, Hillsdale, N.J. Back-propagation: Theory, Architectures and Applications, 433–486 (1995).