A Visionary Way to Novel Process Optimization Techniques
The Transfer of a Process Modeling Language to the Neuronal Level

Norbert Gronau and Marcus Grum
Department of Business Informatics, esp. Processes and Systems, University of Potsdam,
August-Bebel-Strasse 89, 14482 Potsdam, Germany
ngronau@lswi.de, mgrum@lswi.de

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

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.

1 INTRODUCTION

A great potential of Artificial Neural Networks (short: 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 and numerous improvements can be found in late 1980 - 2000 (Schmidhuber, 2015). 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 (Chambers and Mount-Campbell, 2000).

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, the neuronal modeling and decoding of complex processes has not been realized, yet.

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 and optimization world on the one hand and the ANN world on the other hand, such that a sub research question is: "How can a process modeling language be transported on a neuronal level?"

In the following, a *Neuronal Process Modeling* 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* 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 and their difference quantity. The *Neuronal Process Optimization* 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. Within this paper, a focus lays on the Neuronal Process Modeling.

The research approach is intended to be design-oriented as Peffers proposes (Peffers et al., 2006; Peffers et al., 2007), such that the paper is structured as follows: Section 2 presents underlying concepts; Section 3 derives objectives for a Neuronal Process Modeling; Section 4 provides the design, followed by its demonstration (Section 5) and evaluation (Section 6); Finally, Section 7 concludes the paper.

2 UNDERLYING CONCEPTS

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

2.1 Process Optimization

Following the fundamental procedure model for simulation studies of Gronau (2017), 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 results collected, analyzed and interpreted. As changes or optimizations are required, adjustments are defined and simulations tested as long as a sufficient solution has been identified. This will be realized.

The following starts with the understanding of process models to be a homomorphous mapping of a system that reduces the complexity of the real world with respect to the modeling objectives (Gronau, 2016). According to Krallmann et al. (2001), 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 systems, several process modeling languages can be used. Considering organizational, behavior-oriented, informational and knowledge-oriented perspectives, Sultanow et al. (2012) identify the Knowledge Modeling Description Language (short: KMDL) to be superior in the comparison of twelve common modeling approaches.

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 (Remus, 2002). 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 (Gronau and Maasdorp, 2016).

This language has been developed over more than ten years iteratively. Having collected experiences in numerous projects of numerous application areas such as software engineering, product development, quality assurance and investment good sales, the evolution of the KMDL can be found in (Gronau, 2012). Currently, the development of a third version is in progress (Gronau et al., 2016b). In addition to the modeling language, the KMDL reaches a fully developed research method which is described by (Gronau, 2009) in detail.

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, 1995) and 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 (Gronau and Maasdorp, 2016).

Once, a valid process model has been created, a dynamic process can be simulated. Aiming to gain insights within a closed simulation system, the intention is to transfer them 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 can be evaluated quickly and at low costs: current sequences of operations, as well as plans and process alternatives. Those evaluations can be realized before expensive adjustments within current process models are carried out (Gronau, 2017).

2.2 Knowledge Representation

Nonaka and Takeuchi distinguish between explicit knowledge and tacit knowledge (Nonaka and Takeuchi, 1995). While the first can be verbalized and externalized easily, the second is hard to detect. On-building, 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, 1995) seem attractive for the modeling on a neuronal level. Further, the KMDL builds on them, which is selected for demonstration purposes.

2.3 Neuronal Networks

Originally, neural networks were designed as mathematical models to copy the functionality of biological brains. First researches were done by (Rosenblatt, 1963), (Rumelhart et al., 1986) and (McCulloch and Pitts, 1988). 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 (Rosenblatt, 1958), Hopfield Nets (Hopfield, 1982), Multilayer Perceptrons (Rumelhart et al., 1986), (Werbos, 1988), (Bishop, 1995), Radial Basis Function Networks (Broomhead and Lowe, 1988) and Kohonen maps (Kohonen, 1989). 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 a 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 (Plaut et al., 1986) and (Bishop, 1995), PROP (Riedmiller and Braun, 1993), quickprop (Fahlman, 1989), conjugate gradients (Hestenes and Stiefel, 1952), (Shewchuk, 1994), L-BFGS (Byrd et al., 1995), RTRL (Robinson and Fallside, 1987) and BPTT (Williams and Zipser, 1995). As the weight adjustment can be interpreted as a small step in an optimization direction, 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 2.1).

Since neuronal networks model human brains and model the knowledge of a certain learning task, the following refers to neuronal networks as *neuronal knowledge models*.

3 OBJECTIVES OF A NEURONAL PROCESS MODELING

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 following its neurons biological models.

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 A NEURONAL PROCESS MODELING

The following gives definitions of the concept of neuronal modeling. For this, basic definitions are given firstly and on-building definitions 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, 1995), 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 do consider only one neuronal conversion type, while *impure* complex neuronal conversion do 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 participating neurons.

All together, those definitions are the basis for the transfer of a process modeling languages to the neuronal level.

5 DEMONSTRATION OF THE NEURONAL PROCESS MODELING

The following subsections show the realization of the neuronal process modeling on behalf of the *KMDL*. For this, theoretic examples and corresponding process process models are given, that visualize basic definitions. Then, practical examples follow.

5.1 Theoretical Examples

Definitions as they were given in section 4 are visualized in the following three theoretical examples: First, atomic knowledge conversions on a neuronal level can be found in Figure 1.

In this Figure, one can see a neuronal socialization on the top left, a neuronal externalization on the top right, a neuronal combination on the bottom right

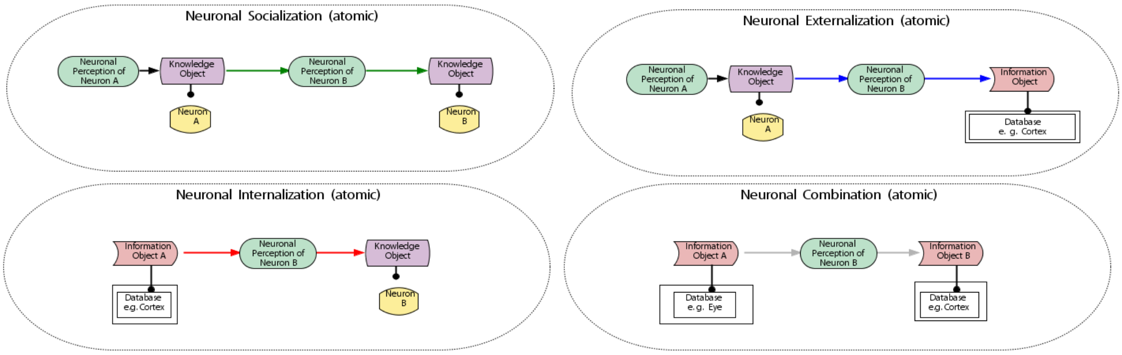

Figure 1: Atomic neuronal conversions.

and a neuronal internalization on 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.

Second, complex neuronal conversions are visualized in Figure 2.

Again, in this Figure, one can see a neuronal socialization on the top left, a neuronal externalization on the top right, a neuronal combination on the bottom right and a neuronal internalization on 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 therefore can be realized with help of the listener concept or the process view.

Third, an abstract neuronal conversion can be found in Figure 3.

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

5.2 Practical Examples

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.

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 or storage areas. Once, they are needed, they can be brought to the corresponding process step with help of transportation elements (Gronau et al., 2016a). 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

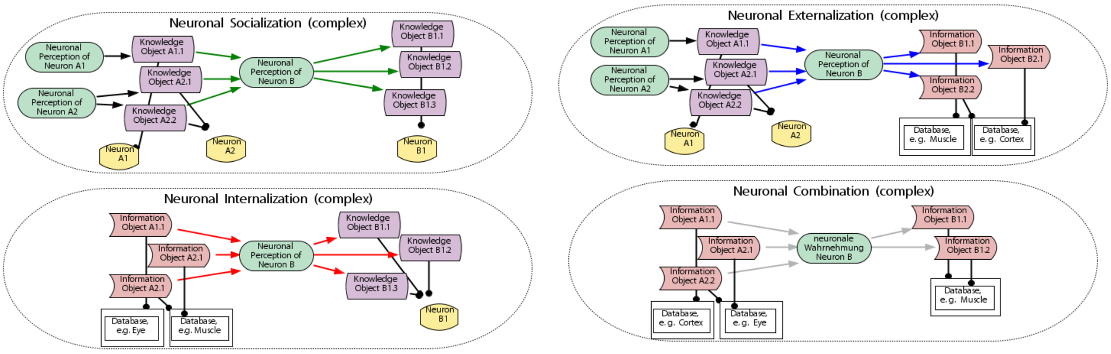

Figure 2: Complex neuronal conversions.

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.

The best choice to realize the entire process model is not always to realize all process parts in 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 sense full to create action alternatives but start fighting a fire immediately like a reflex.

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 as neuronal knowledge models are modeled within the activity view characterizing a certain person. Here, a decomposition 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.

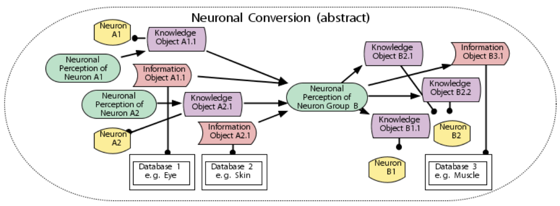

Figure 3: Abstract neuronal conversion.

Further, objectives have been considered coming from a neuronal techniques side as follows: As learning with neuronal networks is not affected by the here presented concepts, neuronal tasks can follow the neurons biological models (objective 6). A parallel neuronal task realization within neuronal networks has been considered (objective 7) as can be seen in Figure 2 (neuronal socialization and neuronal externalization) and Figure 3. 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. 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 Figure 3. 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 for example in Figure 1 and Figure 2. In both Figures, possible sensory information flows can be seen on the bottom (neuronal internalization and neuronal combination). Possible knowledge flows can be seen in both Figures on 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, as can be seen in Figure 1 and Figure 2. 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 the neuronal level, such that a neuronal process modeling, a neuronal process simulation and a neuronal process optimization can be built on base of that.

7 CONCLUSIONS

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 a transfer of a common process modeling language 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 sub research question was answered and the following potentials are suitable next steps:

The concretion of the functioning 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. Still promising is the rebuilding of common process model optimization on behalf of the here presented concepts.

The application of the here presented concepts are assumed to cause a fundamentally 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 is strongly improved. Further, common optimization potentials can be estimated efficiently. Additionally, new optimization approaches and optimization potentials can be identified.

REFERENCES

Bishop, C. (1995). Neural Networks for Pattern Recognition. *Oxford University Press, Inc., ISBN 0198538642*, page 1.

Broomhead, D. and Lowe, D. (1988). Multivariate functional interpolation and adaptive networks. *Complex Systems*, 2:321–355.

Byrd, R. H., Lu, P., Nocedal, J., and Zhu, C. Y. (1995). A limited memory algorithm for bound constrained optimization. *SIAM Journal on Scientific Computing*, 16(6):1190–1208.

Chambers, M. and Mount-Campbell, C. (2000). Process optimization via neural network metamodeling. *International Journal of Production economics*, 79(2000):93–100.

Fahlman, S. (1989). 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*, pages 38–51.

Gronau, N. (2009). *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.

Gronau, N. (2012). *Modeling and Analyzing knowledge intensive business processes with KMDL - Comprehensive insights into theory and practice*. GITO mbH Verlag Berlin.

Gronau, N. (2016). *Geschäftsprozessmanagement in Wirtschaft und Verwaltung*. Gito.

Gronau, N. (2017). *Geschäftsprozessmanagement in Wirtschaft und Verwaltung*, volume 2. Gito.

Gronau, N., Grum, M., and Bender, B. (2016a). Determining the optimal level of autonomy in cyber-physical production systems. *Proceedings of the 14th International Conference on Industrial Informatics (INDIN).*

Gronau, N. and Maasdorp, C. (2016). *Modeling of organizational knowledge and information : analyzing* *knowledge-intensive business processes with KMDL.* GITO mbH Verlag Berlin.

Gronau, N., Thiem, C., Ullrich, A., Vladova, G., and Weber, E. (2016b). Ein Vorschlag zur Modellierung von Wissen in wissensintensiven Geschäftsprozessen. Technical report, University of Potsdam, Department of Business Informatics, esp. Processes and Systems.

Hestenes, M. R. and Stiefel, E. (1952). Methods of conjugate gradients for solving linear systems. *Journal of Research of National Bureau of Standards*, 49(6):409–436.

Hopfield, J. J. (1982). Neural networks and physical systems with emergent collective computational abilities. *PNAS*, 79(8):2554–2558.

Kohonen, T. (1989). Self-organization and associativ memory. *Springer-Verlag New York, Inc., New York, NY, USA, ISBN 0-387-51387-6*, 3rd edition:1.

Krallmann, H., Frank, H., and Gronau, N. (2001). *Systemanalyse im Unternehmen*. Oldenbourg Wissenschaftsverlag.

McCulloch, W. S. and Pitts, W. (1988). A logical calculus of the ideas immanent in nervous activity. *MIT Press, Cambridge, MA, USA, ISBN 0-262-01097-6*, pages 15–27.

Nonaka, I. and Takeuchi, H. (1995). *The knowledge-creating company: How Japanese companies create the dynamics of innovation*. Oxford university press.

Peffers, K., Tuunanen, T., Gengler, C. E., Rossi, M., Hui, W., Virtanen, V., and Bragge, J. (2006). 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.

Peffers, K., Tuunanen, T., Rothenberger, M. A., and Chatterjee, S. (2007). A design science research methodology for information systems research. *Management Informations Systems*, 24(3):45–78.

Plaut, D. C., Nowlan, S. J., and Hinton, G. E. (1986). Experiments on learning backpropagation. *Technical Report CMU-CS-86-126, Carnegie-Mellon University, Pittsburgh, PA*, page 1.

Remus, U. (2002). *Process-oriented knowledge management. Design and modelling.* PhD thesis, University of Regensburg.

Riedmiller, M. and Braun, H. (1993). A direct adaptive method for faster backpropagation learning: The RPROP algorithm. *Proc. of the IEEE Intl. Conf. on Neural Networks, San Francisco, CA*, pages 586–591.

Robinson, A. J. and Fallside, F. (1987). The utility driven dynamic error propagation network. *Technical Report CUED/F-INFENG/TR.1, Cambridge University Engineering Department*, page 1.

Rosenblatt, F. (1958). The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, 65:386–408.

Rosenblatt, F. (1963). Principles of Neurodynamics. *Spartan, New York*, page 1.

Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1986). Learning internal representations by error propagation. *MIT Press, Cabridge, MA, USA, ISBN 0-262-68053-X*, pages 318–362.

Schmidhuber, J. (2015). Deep learning in neural networks: An overview. *Neural Networks*, 61:85 – 117.

Shewchuk, J. R. (1994). An introduction to the conjugate gradient method without the agonizing pain. *Technical Report, Carnegie Mellon university, Pittsburgh, PA, USA*, page 1.

Sultanow, E., Zhou, X., Gronau, N., and Cox, S. (2012). Modeling of processes, systems and knowledge: a multi-dimensional comparison of 13 chosen methods. *International Review on Computers and Software (IRECOS)*, (6):3309–3319.

Werbos, P. J. (1988). Generalization of backpropagation with application to a recurrent gas market model. *Neural Networks*, page 1.

Williams, R. J. and Zipser, D. (1995). 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*, pages 433–486.