Kannisto, P., Hästbacka, D. & Kuikka, S. (2017).
System Architecture for Mastering Machine Parameter Optimisation.
Computers in Industry 85, pages 39-47
DOI: 10.1016/j.compind.2016.12.006

System Architecture for Mastering Machine Parameter Optimisation

Petri Kannisto*, David Hästbacka, Seppo Kuikka

Tampere University of Technology

Department of Automation Science and Engineering

P.O. Box 692, FI-33101 TAMPERE, Finland

{*petri.kannisto*|*david.hastbacka*|*seppo.kuikka*}*@tut.fi*

Abstract

In mobile machines, as well as in manufacturing, the overall productivity is essential for business competitiveness. As the operation of a modern mobile machine is affected by various parameters, they need to be tuned to reach an optimal performance – however, due to machine complexity, parameter optimisation is difficult for a typical operator. To enable parameter optimisation locally in machines, this article presents a system architecture to generate information and knowledge from machine fleet data and to utilise them in machine operations in the field. Measurement data is collected and analysed to discover the associations between machine performance and parameter values. While some results are plain statistical distributions, any resulting more sophisticated domain knowledge is stored as rules. Rule-based reasoning enables a zone of interoperation between the information system and domain experts. Once information and knowledge have been generated, they are made available to machines that run the actual parameter assessment application. Results made with forestry data indicate that the system has a considerable potential to improve machine productivity.

Keywords: Distributed Systems, Knowledge Management, Performance Optimization, Mobile Machines

*Corresponding author

1. Introduction

Even a minor productivity improvement may result in a substantial advantage as well as reduced fuel consumption and overall environmental impact. In material handling and processing, mobile machines are essential in the production chain. While modern equipment has a high automation degree, there is still room for significant improvement in its operation.

Mobile work machines are complex technical systems that consist of a number of *parametrisable* components. Typically, components are customisable rather than having a fixed operating context. For example, while the typical excavator task is to dig, different soil types may require different parameters for the optimal performance. Further, even work types may vary (such as pile driving instead of digging).

Mastering parameter knowledge to optimise performance is not a trivial task in a large scale. The number of parameters may reach hundreds in a modern machine. While a domain expert might have a solid basic understanding of typically good parameter values, an average operator certainly has not. Besides, even domain experts do not know all parameter–performance dependencies so advanced analysis methods are required to discover new knowledge from measurement data. The ultimate goal is to raise the automation degree of parameter optimisation: how to take control over a large data set collected from a machine fleet, how to manage parameter and performance value knowledge and how to apply it in individual machines during operation. Presumably, such optimisation is extensively performed autonomously by machines in the future. In this paper, a distributed system architecture is introduced for the task.

The research methodology followed by the work is to resolve requirements, to create a concept based on them and to evaluate the concept by implementing a prototype. The methodology can be seen to follow both design science and constructive research approach as presented by Piirainen & Gonzalez [1]. Section 2 summarises related work. The various aspects and requirements of the system are covered in section 3. The design of the system is introduced in section 4 and a prototype is presented in section 5. Finally, section 6 contains results and discussion followed by conclusions in section 7.

2. Related Work

This section covers previous studies related to industrial service architectures. Also, vehicle data collection and refinement as well as rule based systems are considered.

Bringing service-oriented design to the industrial context has been discussed in various studies. Jardim-Goncalves et al. propose a platform to improve enterprise collaboration and system interoperability in industry [2]. Colombo & Karnouskos argue that service design eases both device integration and reconfiguration after constantly changing business requirements [3]. Cândido et al. propose an infrastructure where services facilitate device deployment during production system lifecycle [4]. The E-maintenance concept in manufacturing includes not only equipment data collection and utilisation but also knowledge management for decision support. Bangemann et al. write about a maintenance systems integration platform that enables geographical distribution and utilises web services [5]. Karim proposes a framework for service-oriented E-maintenance applications as well as a methodology for the identification of supportive services [6]. A practical industrial service architecture for condition monitoring has been introduced by Hästbacka et al. [7].

Vehicle and machinery data utilisation have been studied in several articles. Lu et al. have researched fuzzy rules generation for fault diagnostics [8]. Dingus et al. have documented the collection of a large car data set in everyday conditions [9]. Wu et al. have utilised mathematical methods to recognise engine faults from audio data [10]. Palmroth has researched the analysis of mobile machine data to assist operator learning [11]. Golparvar-Fard et al. have developed an algorithm to recognise earthmoving equipment actions from video [12].

He et al. demonstrate how a cloud application may assist either in finding a parking place or in vehicle data mining [13].

Particularly in agriculture, machinery data acquisition is a growing concern. Steinberger et al. as well as Peets et al. have studied data collection from heterogeneous data sources [14, 15]. Iftikhar & Petersen have researched bidirectional data transfer from and to machinery [16]. Fountas et al. have designed a farm machinery information system to facilitate data utilisation [17].

Rule based systems are useful tools in the energy management of hybrid vehicles and machines. For example, Lin et al. have utilised dynamic programming to generate rules for power management [18]. Hybrid excavators have been included in power management research by, for instance, Kim et al. [19].

Rules have also other applications in mobile machines. Rules can assist in selecting the best machine or equipment for some specific purpose as studied by, for example, Amirkhanian & Baker [20]. Further, den Hartog et al. have utilised rule based models to predict the performance of mobile machines [21]. Bradley & Seward have utilised rules to raise the intelligence and the automation degree of excavation [22].

Compared to previous research, the work to be presented is unique as it combines the aspects of a service architecture, a distributed machine fleet, machine data refinement and refined data utilisation locally in machines as well as knowledge management with rules. Two previous works have considered system architecture, machine data processing and rules for knowledge management. Kannisto et al. have developed an architecture for operator feedback generation [23]. Kannisto et al. introduce a system for mobile machine parameter optimisation [24]. Compared to this work, it is on a more conceptual level, and no experimental results are presented.

3. Machine Information Management Requirements

Parameter optimisation is likely present wherever mobile machines are utilised. While this section considers the problem from a general viewpoint, forestry domain is also considered particularly. This work is considered novel as equally comprehensive publications about service architectures for managing machine parameter optimisation are not known to exist.

3.1. Domain Related Challenges as Motivator

An information system to aid parameter optimisation would be beneficial in forestry machines. They have several instruments for handling tree stems; for instance, the boom of a machine has typically several actuators that operate boom joints or grab or process stems. While operator skills are important, the overall machine performance is largely affected by the precision and speed the machine responds to operator actions which is effectively determined by machine parameters. (The influence of machine parameters has been suggested by Väyrynen et al. [25].) Unfortunately, parameter optimisation requires knowledge that is unavailable for typical machine operators, and even domain experts do not have all the knowledge potentially available in measured machine data. The global or even regional variety of forests – and its effect on which parameter values perform best – brings additional challenge so operating contexts should also be considered. In the end, not only data analyses are required but their results should also be available for operators to assist parameter optimisation. Further, domain information and knowledge are expected to evolve constantly as new data is collected and new analyses are executed – that is, repeated updates are required. The parameter optimisation process can be illustrated with a loop as in Figure 1.

The solution should enable the generation, distribution and exploitation of domain information and knowledge discovered in data analyses – even local utilisation in machines is desirable. Depending on the industry, considerable requirements and limitations may arise from machine distribution. In forestry, machines may operate far from each other, the corporate office and public infrastructure. Also, the machines may have no Internet connectivity for days or even weeks. In a business ecosystem, parameter optimisation may be managed by various actors (at least operators, the machine manufacturer or a local

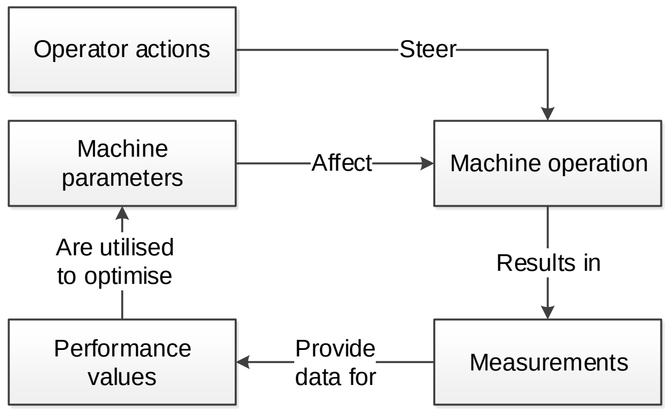

Figure 1: Parameter optimisation loop for improving the overall machine performance.

dealer). Their scope and access to physical machines varies, but distribution is inevitable.

3.2. Required System Features

The research questions are as follows:

> What kind of conceptual information system architecture is required to centrally manage the information and the knowledge related to machine parameter optimisation? How to enable the distribution of information and knowledge to geographically dispersed machines so they can be utilised locally during operation? How to implement such an information system?

To concretise parameter optimisation, let us look at an example about determining the amount of hydraulic flow directed to a machine boom. The flow basically determines the power available for boom operations: more flow results in faster responses. However, at some point, the motion would become even too quick causing inaccuracy and controlling difficulties. That is, the goal is to balance power and accuracy (see Figure 2).

The simplest form of parameter optimisation is performed using two types of information: relative performance and relative parameter values. First, the

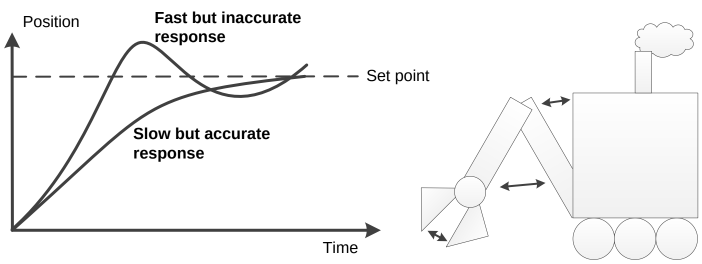

Figure 2: Unsuitable parameter values may result in too slow or too inaccurate responses to operator actions.

absolute performance of the machine is determined from measurements. Then, by comparing to past performance values, relative performance is determined. If performance is low, relative parameter values are generated by comparing absolute parameters to ones that have provided a good performance in the past. A simple feedback is to suggest to use historical well-performer parameters.

As the parameter optimisation performed locally in machines utilises fleet-wide information, a significant interaction requirement appears. To have fleet scope results, large data analyses must be performed in the corporate office. Then, to enable the utilisation of the results in each machine, they must be delivered and cached locally as no persistent Internet connectivity can be assumed.

The diversity in operating environments causes systematic variation in performance and parameters so some context classification method is required. Each domain has its own context characteristics: forests and trees in forestry, soil types in excavation, fields and plants in agriculture and so forth. Differences in performance values are expected, and even parameter value adaptation may be required. Whatever are the method and resolution in context classification, consistency in data analyses is guaranteed by using identical methods in the corporate office and in each machine. Moreover, if the context classification method evolves, an update mechanism is required.

Additional complexity is caused by domain knowledge utilisation. There are likely cases where domain experts know some "rules of thumb" that may

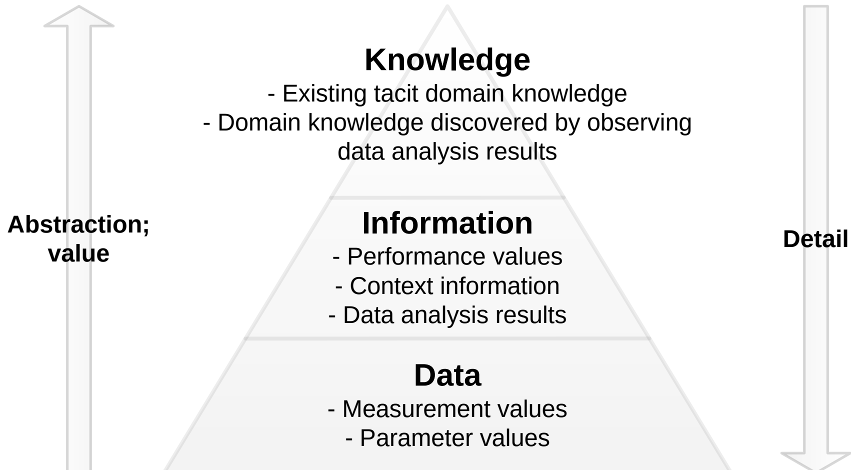

Figure 3: Processing detailed data enables refinement to generate of information and knowledge; more value is gained by raising the abstraction level.

consist of arbitrary condition structures on performance and parameter values. The support for such knowledge representation likely adds to adaptability and the number of use cases in parameter optimisation. As domain experts are rarely ICT experts, an easy-to-use interface should exist for knowledge management. Another requirement is that whichever implementation technology is chosen, it must be possible to replace the domain knowledge modelling environment without otherwise re-engineering the system (ease of deployment is desirable in system evolution). Finally, even domain knowledge must be delivered to machines for local utilisation.

To create a structure for information processing, the various information types have been positioned in the data–information–knowledge triangle (referred to, for instance, by Ackoff [26] as cited by Rowley [27]). Figure 3 illustrates this: data consists of raw values while information is refined from it, and knowledge covers more complex domain expertise. While the information level is most essential in feedback generation, the knowledge level brings additional adaptability.

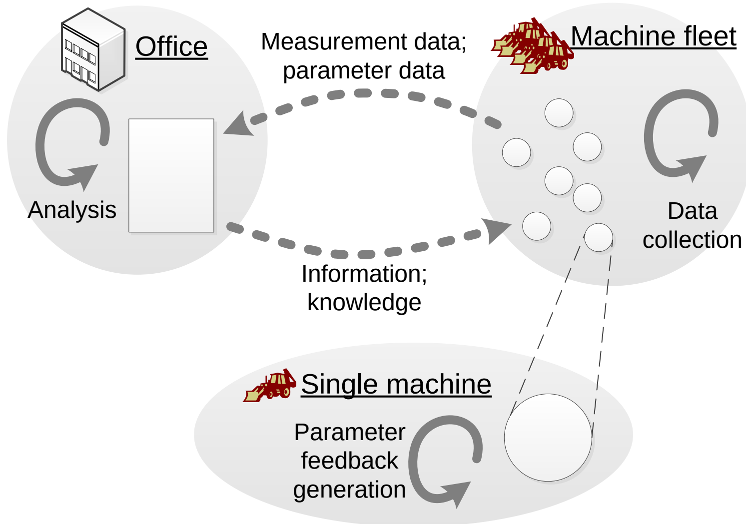

Figure 4: Machine data is collected for analysis. With the resulting information and knowledge, parameter value feedback generation is enabled in machines.

The requirements are summarised in Figure 4. The figure emphasises distribution requirements: fleet data is collected for centralised management and analysis, and its results are delivered back to machines for local utilisation. The concept has an asynchronous nature: although there are links to transfer information between instances, machines operate independently of each other and the office.

4. Parameter Optimisation Architecture

In conformance with the requirements, this section specifies a conceptual system architecture. As the ultimate requirement is feedback generation locally in machines, its functionality is explained first. Then, the required supportive architectural features are discussed.

4.1. Local Feedback Generation in Machines

The feedback generation flow executed in machines is illustrated in Figure 5: it includes context recognition, the generation of relative performance and pa-

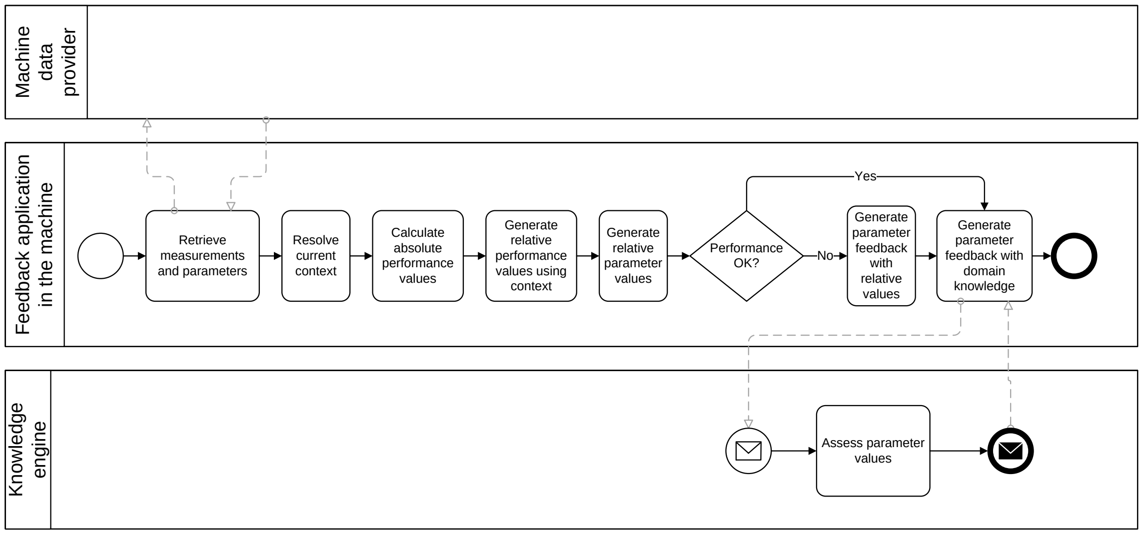

Figure 5: The flow of generating parameter feedback within a single machine for the operator.

rameter values and, finally, the application of domain knowledge. First, machine data is retrieved after which the prevailing context class is resolved. Then, relative performance values are generated using fleet-wide aggregated information. If machine performance is insufficient, a simple parameter value comparison is executed. Finally, domain knowledge is utilised for further parameter optimisation – this is always performed as parameter tuning might be desirable even if overall performance seemed fine. In the hydraulic boom power optimisation example, boom performance could be fine even if fuel consumption were weak.

Context information is utilised in the generation of relative parameter and performance values. Context classes have been defined in data analysis – for each context class, there is a dedicated set of aggregated performance and parameter values. What is observed in context classification depends on the domain where the machine operates (forestry, earthmoving, agriculture and so forth) and the type of work being performed – anything that affects performance values should be considered. Figure 6 illustrates relative value generation – on a high level, this applies to both parameters and performance.

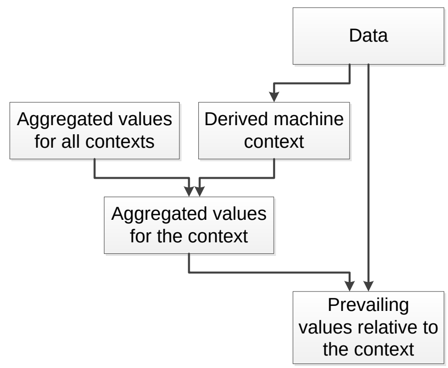

Figure 6: How operating context is considered in the generation of relative values.

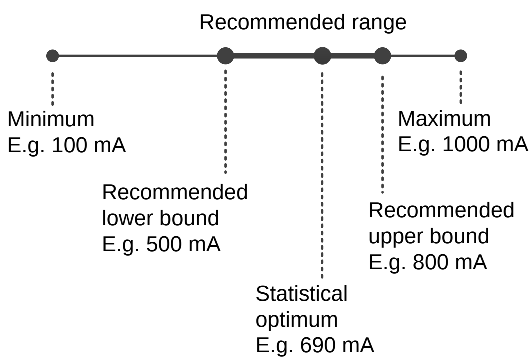

Figure 7: An illustration of a parameter range.

Relative parameter values are generated by comparing numeric values to *ranges* associated to good performance in the identified context in the past. In Figure 7, the parameter value represents the control current of a hydraulic actuator. In the related operating context, the recommended range is from 500 mA to 800 mA. Statistically, the best performance has been reached with 690 mA. In the hydraulic boom example, such a parameter could determine the positioning current of the valve that determines the flow (or power) directed to the boom.

Relative performance values are generated from distributions. For each performance measure, there will be a distribution in each context class. Relative values are represented as percentiles: for example, a value higher than 75% of past values has percentile 0.75. The value does not indicate if high is good or not – for example, high productivity is desirable while high fuel consumption is not.

4.2. Domain Knowledge Representation

Domain knowledge is represented as *rules*. Whichever the chosen implementation framework or technology is, the previously given requirement of easy knowledge modelling sets some restrictions to be considered.

To demonstrate a rule, let us again look at the hydraulic boom example. The following pseudocode represents a rule about lowering the parameter determining the amount of flow (or power) the hydraulic boom receives. The rule is necessary if the logic is not reached with the regular relative parameter value comparison.

```
IF
  Measure "Productivity" is "weaker than average" AND
  Measure "Corrective manoeuvres" is "weaker than average" AND
  Parameter "Fluid to boom" is "above optimum"
THEN
  Lower parameter "Fluid to boom"
```

In knowledge modelling, fuzzy values are utilised instead of numbers to enable more power of expression. The high resolution enabled by percentiles may bring only a little value as uncertainty is always present in measurements. So, to put weight on expression, elaborated fuzzy values are utilised. Here, a percentile value falls to one of four slots: it can be *weak*, *weaker than average*, *better than average* or *good* (complements of these being considered as well); see Figure 8. It must be noted that whether high or low is good is also considered here – thus, it must be defined for each measure (a high productivity is good while a high fuel consumption is not). For instance, a configuration file or a rule set may be utilised for the definition.

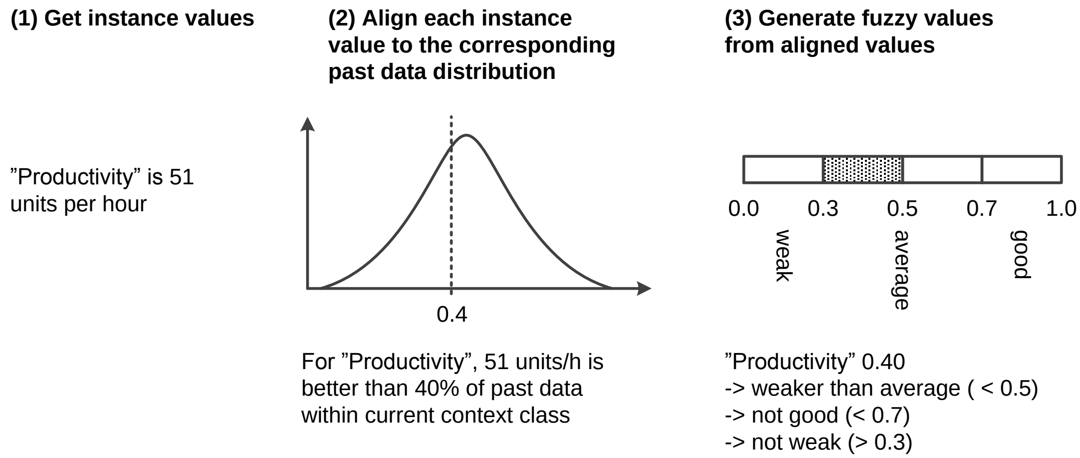

Figure 8: The generation of fuzzy values from measurement data.

4.3. Conceptual Architecture and Design Principles

Considering previously introduced aspects, a conceptual architecture of data flows and refinement can be created (see Figure 9). The application logic in the machine may be run during machine operation while office operations will be performed periodically without direct connection to machines. The degree of automation is lower in the office as both human effort and critical human inspection are required in data analysis. Data is collected from a large machine fleet to a central storage – a cache is utilised in each machine as no persistent Internet connectivity can be assumed. Data analyses reveal what kind of performance results from various parameter value combinations in each context class (as in the work of Väyrynen et al. [25]). The resulting information about those associations is exploited locally in each machine; it is utilised to determine the prevailing relative performance and parameter values. Naturally, a context classification similar to the one in the office must be utilised. Due to Internet connectivity limitations, caching is applied to the office supplied information in machines. For domain knowledge, a repository is held in the office, and a cache exists in machines.

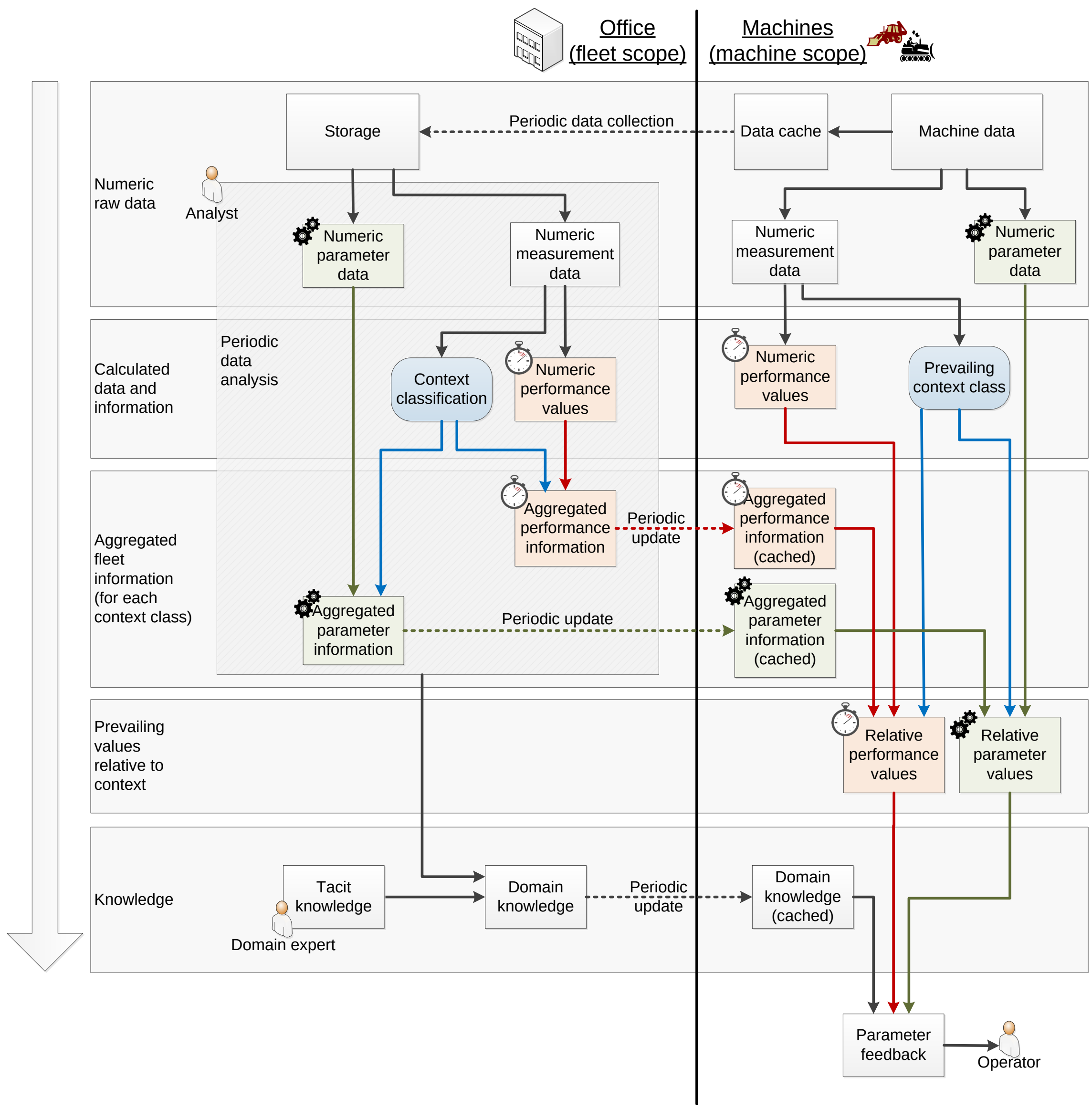

Figure 9: Data is collected from machine fleet after which analyses can be performed in the office. The resulting information and knowledge are delivered to machines. Finally, using the information, the knowledge and local machine data, parameter value feedback is generated locally in machines. As more data is collected during the system lifecycle, new analyses are performed periodically in the office.

Open platform-independent technologies as well as loose coupling are favoured in component interface implementation. First, the goal is to minimise any dependencies to a specific platform. As the system consists of multiple components, a need to replace some of them may occur during its lifecycle. It is beneficial that no platforms are excluded by design as machine control and information system platforms vary and evolve. Second, loose coupling will make component replacement easier. Replacing one component should not require any changes in other components as long as their interfaces remain the same. This was required explicitly for knowledge modelling as rule frameworks often use proprietary technologies.

5. Parameter Feedback Prototype

Considering the requirements and designed architecture, a prototype has been implemented (see Figure 10). There is a strict division into office environment and machine environment components. The office has a component for rule modelling, and analysts perform data analyses to have aggregated fleet information. As data collection is rather a straightforward function to implement (though requiring additional effort), it has been left as a future task. Parameter ranges and performance distributions are currently stored in plain files. In machines, there are components for feedback generation, machine data retrieval and rule execution. Each software component is explained in detail in the following paragraphs.

The prototype has been designed to run in a forestry machine that provides a service interface for data retrieval. The interface utilises XML (Extensible Markup Language) and HTTP (Hypertext Transfer Protocol) so integration is easy per a wide support in software libraries.

The actual feedback generation is performed by the *feedback engine* component. It utilises local machine data, cached parameter ranges and performance distributions, and the rule executor component to generate parameter feedback.

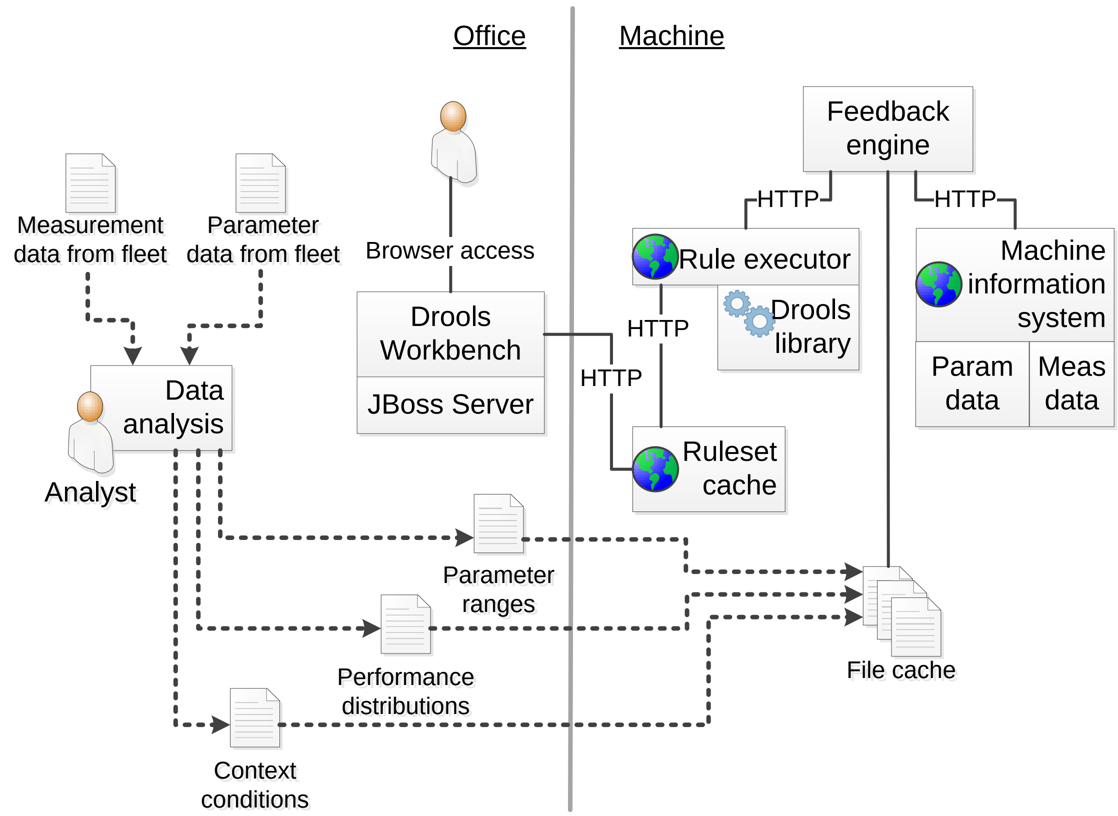

Figure 10: The office and the machine environment components of the implemented prototype.

For domain knowledge, there is local file caching. In ruleset retrieval, the ruleset cache is the active component while the office simply provides it for download. The ruleset cache has a service interface so it is loosely coupled to its consumers.

Domain knowledge is represented as rules and modelled with the Drools framework. While it provides several ways for rule modelling, decision tables are utilised as they require only basic ICT knowledge from the rule modeller. The rule executor service performs domain knowledge based inference in machines. The service has a loosely coupled platform independent interface so even a complete substitution is possible – in that case, no service consumer re-engineering is required. Below, there is an example of representing a productivity value in the format of the interface: object types and their properties are not predefined. The format can be mapped for utilisation in any object based interaction.

- Object type = "**Measure**"
  - Property name = "**name**" value = "**Productivity**"
  - Property name = "**value**" value = "**Weaker than average**"

In the current prototype version, parameter range and performance distribution delivery from office to machines has not been implemented, but dedicated XML based formats have been specified. The XML documents are currently delivered manually. With modern technologies, document retrieval from a web server is a straightforward task, but the required setup was considered to bring only little additional value in the prototype. An implementation similar to domain knowledge caching would presumably suit here as well.

Contexts are recognised by utilising a configurable tree stem classification condition set in the prototype. Stems are classified after their properties, and the predominant class is considered the context – the objective is to restrict processing to stems that are comparable to each other. While the context recognition logic is simple, the prototype enables its configuration as the conditions are parsed from a text file. Practically, any operating environment properties available from the machine information system may be utilised. To keep data utilisation consistent, the conditions should be generated along with the data analysis that generates performance distributions and parameter ranges. While delivered manually, the context definition document could as well be retrieved from a corporate web server.

5.1. Test Setup 1: Operator Training Simulator

The first practical experiments were run in a forestry machine simulator utilised for operator training. The scenario was to optimise the parameters that affect automatic tree stem positioning in a wood processing implement. The function is completely automatic so parameters have a high influence on performance. The simulator had a machine data provider interface identical to a real machine.

The setup had a few limitations related to context recognition and the utilisation of data and domain knowledge. First, for simplification, context recognition was not considered. Second, performance distributions and parameter ranges from physical machines could not be utilised as the simulator physics model was not completely realistic. The initial task was to resolve appropriate performance and parameter values for automatic stem positioning and to create the corresponding distributions and ranges. Third, any rules applied did not contain any complex domain knowledge but they were redundant to other parameter estimation functions.

5.2. Test Setup 2: Real Machine Data

The prototype was also tested with real data captured from forestry machines during operation. Earlier, data analyses had been run on past data to generate appropriate performance distributions, recommended parameter ranges and context recognition conditions. The prototype was run in a plain PC as there was no possibility to run in forestry machines – that way, it was also easy to

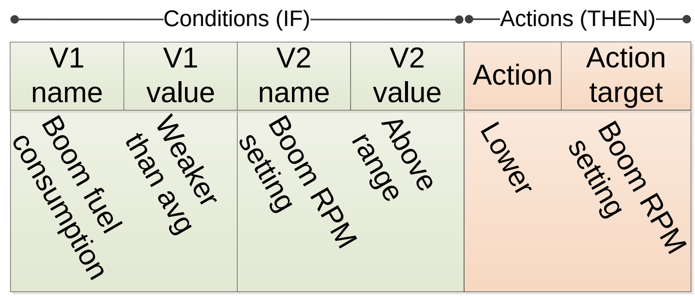

Figure 11: Engine running speed during boom operation rule as a decision table row.

utilise data from multiple machines. However, data was retrieved from an interface identical to a physical machine.

The experiment covered both simple parameter range checks as well as domain knowledge utilisation. The first point was to generate feedback about the parameters affecting automatic tree stem positioning in a wood processing implement. The focus was to test the assessment if parameters are inside the recommended ranges – that is, utilising previously refined performance and parameter information. The second point was to test how knowledge (i.e. rules) utilisation works. Even if some performance measures were good, a machine may still work non-optimally. Here, we looked at the diesel engine RPM (running speed) setting during boom operation as a high value may waste fuel. However, a skilled operator may exploit the additional power of high RPM (thus providing more effective operation) so a rule was created to observe both fuel consumption and the RPM setting (see Figure 11).

6. Results and Discussion

A system architecture has been designed to master data, information and knowledge from a mobile machine fleet so machine parameters can be optimised locally in machines. The system must be aware of various operating contexts, covering environment and type of work. These requirements are met. The system enables data collection and exploitation so information and knowledge may be generated and made available for machines for download. Also, provided that context classes have been considered in data analysis, the appropriate aggregated parameter and performance information set is considered while generating parameter feedback.

In conformance with the requirements of the system, a prototype has been implemented. While the implementation of some system parts remains a future task (raw machine data collection as well as the distribution of parameter ranges, performance distributions and the context classification definition), the business logic was already experimented successfully in two environments. Easy knowledge modelling is enabled by decision tables that operate on fuzzy values. The prototype utilises a machine data retrieval interface identical to that on physical machines so it could be used even during real machine operation. Open platform independent interface technologies and formats have been utilised to promote easy component deployment during system evolution.

First, a short test in a realistic operator training simulator demonstrated how non-optimal parameter values resulted in a low performance and appropriate feedback. The application utilised performance distributions to determine relative performance and recommended parameter ranges to indicate how parameter values should be changed. Parameters were also estimated with rules. As simulator parameters had been set to non-optimal values, causing weak machine performance, the feedback application suggested parameter tuning (see Table 1) as expected. A clear shortcoming is that the utilised parameter ranges and performance distributions were not from real operative data but from the simulator; still, their utilisation was similar as if retrieved from a fleet-wide data container server. Further, there was no actual case for domain knowledge utilisation; instead, the rules were redundant to other functionality. In addition, context recognition was not utilised at all. Finally, although data coverage was relatively low here, appropriate output was generated during operation, and a system interface identical to a physical machine was utilised.

Second, a more comprehensive test with real machine operation data was performed. A total of 11 machines was included. The data of each machine covered the processing of at least 1000 tree stems. The results are shown in

Table 1: The prototype was tested in a realistic forestry machine simulator to observe automatic stem positioning parameters and the resulting performance. In each test run, automatic positioning was performed at least 20 times. The parameters had no effect on *positioning error*. However, *positioning time* deteriorated when parameter values were non-optimal. As performance was weak (underlined), the system suggested to set the parameters (bold) to their statistically optimal values (by both plain parameter value comparison and rule-based assessment).

Run	Description	Measure Positioning error	Measure Positioning time	Parameter Max speed	Parameter Approach speed	Parameter Approach distance
1	Optimal approach	Good	-	Optimal	Optimal	Optimal
2	Quick and short approach	Good	-	>Optimal	>Optimal	<Optimal
3	Optimal approach	-	Good	Optimal	Optimal	Optimal
4	Quick and short approach	-	Weak	**>Optimal**	**>Optimal**	**<Optimal**
5	Short approach	-	Weak	**>Optimal**	Optimal	**<Optimal**

Table 2. Machines 4 and 5 had a good performance so no parameter feedback was given. Machine 3 had a bad performance but it could not be explained with parameters. All the other machines had at least one bad positioning parameter value. A high RPM during boom usage setting does not seem to explain high fuel consumption in boom utilisation as the RPM lowering rule only fired for machine 2 although 6, 8, 9 and 10 also have a high boom fuel consumption.

The second test run complements the coverage of the first. Operating data sets from multiple physical machines were utilised, and performance distributions as well as parameter ranges had also been generated from real operational data. Each data set is relatively large, and the RPM rule demonstrated knowledge modelling. However, the analyses were only run afterwards and not during machine operation. Still, from the architectural point of view, the setup was close to realistic as machine data was retrieved from an interface identical to real machines.

Table 2: Parameter optimisation results with real machine operation data.

Mach. ID	Positioning performance	Positioning parameters out of range	Boom fuel consumption	Lower boom RPM suggested
1	Weak	Approach speed	Better than avg	No
2	Weaker than avg	Approach speed	Weaker than avg	**Yes**
3	Weak	(None)	Better than avg	No
4	Good	(Not estimated)	Better than avg	No
5	Good	(Not estimated)	Better than avg	No
6	Weaker than avg	Approach speed, approach distance	Weaker than avg	No
7	Weak	Approach speed	Good	No
8	Weaker than avg	Approach speed	Weaker than avg	No
9	Weaker than avg	Approach speed, max speed	Weak	No
10	Weak	Approach speed, max speed	Weaker than avg	No
11	Weaker than avg	Approach speed, max speed	Good	No

The context recognition method utilised in the second test run appeared ineffective. It classified stems only after their diameters, and the class with most stems was declared the context. Surprisingly, the same class was determined for each test run. That is, method should be considered carefully as it seemed to provide little practical value.

All in all, the prototype has concretised the functionality of the concept in practical experiments. The first experiment in a simulator demonstrated parameter feedback generation right after operation while the second experiment demonstrated real fleet-wide machine data utilisation. In both the experiments, the prototype discovered non-optimal parameter values and gave feedback to change them to improve machine performance.

The system has also room for more advanced design. Especially in forestry, the possibilities of context recognition are wide as there is a huge variation in forests even regionally. With contexts, if comprehensive data sets are collected, a good accuracy may be reached in parameter optimisation – still, the most significant advantage would come as soon as the clearest errors were eliminated. Further, in knowledge representation, the utilisation of rules gives a huge potential. A graph based rule modelling tool would make rule modelling even easier and give more freedom of expression. From the practical point of view, effort should be put on the actual delivery of data, information and knowledge between data analysts and machines. Implementing caching, servers and all the channels required for the traffic is not trivial.

As the potential of the concept has been shown, the various actors in industry should exploit it. Considerable benefits are expected: a better efficiency helps in raising productivity, gaining competitive advantage as well as lowering resources consumption and environmental impact. Software, information and knowledge can be utilised to get more from physical equipment capabilities. Although only experimented in forestry, the architecture and methods are applicable to improve machine performance and efficiency in any domain – such as agriculture or construction. Combined with constantly advancing big data processing, there is potential to develop even completely new optimisation methods following the results of this work. It is expected that computer-assisted or even automatic optimisation becomes a common feature in future machines.

7. Conclusion

In this article, a system architecture for mastering the information and the knowledge required for mobile machine parameter optimisation is presented. Data is gathered from a large machine fleet for analysis so new cause-and-effect information and knowledge about parameters and performance are generated. Both information and knowledge are made available to machines so they can be exploited during operation. As the information and knowledge are under constant evolution, it must be possible to retrieve an up-to-date version to machines when connectivity is available.

In the office environment, there are storages for performance and parameter information as well as domain knowledge. Performance information is stored as distributions while parameter information is represented as ranges of recommended values. Domain knowledge is stored as rules – decision tables are utilised as they require only basic ICT expertise from the rule modeller enabling domain experts to edit them. Operating contexts are also considered. As a machine may operate in various environments and perform various types of work, variation in optimal parameter values as well as performance measures is expected. This requirement is considered by associating performance and parameter information to context classes.

Caching must be utilised in machines so they can submit their data to the central storage whenever Internet connectivity is available. Similarly, each machine must have a cache for the most recent information and knowledge sets retrieved from the corporate office.

In each machine, there is an application to provide parameter feedback to the operator. Utilising local parameter and performance values and the information and the knowledge retrieved from the office, the application assesses prevailing performance and parameter values and suggest parameter adjustment if needed.

In conformance with the design, a prototype has been implemented. Data analysis results and rules are delivered from the office environment to machines. At runtime, parameter and measurement values are retrieved from machine information system to generate feedback. As machine parameters and performance were considered in tests, the prototype proved to be successful by providing appropriate parameter tuning suggestions.

Various future research tasks remain. Currently, the system does not consider the individual characteristics of machines and their components. The system should save state information to detect if parameter value adjustment has actually decreased performance. Further, parameter adjustment should be automatised so the operator would not have to perform it manually.

Acknowledgments

This work was made as a part of the D2I (Data to Intelligence) project funded by Tekes (the Finnish Funding Agency for Innovation). The authors would like to express their sincere gratitude to the project partners and participant companies.

References

[1] K. A. Piirainen, R. A. Gonzalez, Seeking constructive synergy: Design science and the constructive research approach, in: J. vom Brocke, R. Hekkala, S. Ram, M. Rossi (Eds.), Design Science at the Intersection of Physical and Virtual Design, Vol. 7939 of Lecture Notes in Computer Science, Springer Berlin Heidelberg, 2013, pp. 59–72. doi:10.1007/978-3-642-38827-9_5. URL http://dx.doi.org/10.1007/978-3-642-38827-9_5

[2] R. Jardim-Goncalves, A. Grilo, A. Steiger-Garcao, Challenging the interoperability between computers in industry with MDA and SOA, Computers in Industry 57 (89) (2006) 679 – 689, collaborative Environments for Concurrent Engineering Special Issue.

doi:http://dx.doi.org/10.1016/j.compind.2006.04.013. URL http://www.sciencedirect.com/science/article/pii/S0166361506000753

[3] A. W. Colombo, S. Karnouskos, Towards the factory of the future: A service-oriented cross-layer infrastructure, in: ICT Shaping the World: A Scientific View, John Wiley and Sons, 2009, pp. 65–81.

[4] G. Candido, A. Colombo, J. Barata, F. Jammes, Service-oriented infrastructure to support the deployment of evolvable production systems, Industrial Informatics, IEEE Transactions on 7 (4) (2011) 759–767. doi:10.1109/TII.2011.2166779.

[5] T. Bangemann, X. Rebeuf, D. Reboul, A. Schulze, J. Szymanski, J.-P. Thomesse, M. Thron, N. Zerhouni, PROTEUS – creating distributed maintenance systems through an integration platform, Computers in Industry 57 (6) (2006) 539 – 551, e-maintenance Special Issue. doi:http://dx.doi.org/10.1016/j.compind.2006.02.018. URL http://www.sciencedirect.com/science/article/pii/S0166361506000601

[6] R. Karim, A service-oriented approach to e-maintenance of complex technical systems, Ph.D. thesis, Division of Operation and Maintenance Engineering, Luleå University of Technology (2008).

[7] D. Hästbacka, P. Kannisto, S. Kuikka, Business process modeling and SOA in industrial O&M application development, in: Proceedings of the 13th International Conference on Enterprise Information Systems, 2011, pp. 277–285. doi:10.5220/0003507202770285.

[8] Y. Lu, T. Q. Chen, B. Hamilton, A fuzzy system for automotive fault diagnosis: fast rule generation and self-tuning, Vehicular Technology, IEEE Transactions on 49 (2) (2000) 651–660. doi:10.1109/25.832997.

[9] T. A. Dingus, S. Klauer, V. Neale, A. Petersen, S. Lee, J. Sudweeks, M. Perez, J. Hankey, D. Ramsey, S. Gupta, et al., The 100-car naturalistic driving study: Phase II-results of the 100-car field experiment, Tech. rep. (2006).

[10] J.-D. Wu, P.-H. Chiang, Y.-W. Chang, Y. jung Shiao, An expert system for fault diagnosis in internal combustion engines using probability neural network, Expert Systems with Applications 34 (4) (2008) 2704 – 2713. doi:http://dx.doi.org/10.1016/j.eswa.2007.05.010.
URL http://www.sciencedirect.com/science/article/pii/S0957417407001625

[11] L. Palmroth, Performance monitoring and operator assistance systems in mobile machines, Ph.D. thesis, Department of Automation Science and Engineering, Tampere University of Technology, Tampere, Finland (2011).

[12] M. Golparvar-Fard, A. Heydarian, J. C. Niebles, Vision-based action recognition of earthmoving equipment using spatio-temporal features and support vector machine classifiers, Advanced Engineering Informatics 27 (4) (2013) 652 – 663. doi:http://dx.doi.org/10.1016/j.aei.2013.09.001.

[13] W. He, G. Yan, L. D. Xu, Developing vehicular data cloud services in the iot environment, Industrial Informatics, IEEE Transactions on 10 (2) (2014) 1587–1595. doi:10.1109/TII.2014.2299233.

[14] G. Steinberger, M. Rothmund, H. Auernhammer, Mobile farm equipment as a data source in an agricultural service architecture, Computers and Electronics in Agriculture 65 (2) (2009) 238 – 246. doi:http://dx.doi.org/10.1016/j.compag.2008.10.005.

[15] S. Peets, A. M. Mouazen, K. Blackburn, B. Kuang, J. Wiebensohn, Methods and procedures for automatic collection and management of data acquired from on-the-go sensors with application to on-the-go soil sensors, Computers and Electronics in Agriculture 81 (2012) 104 – 112. doi:http://dx.doi.org/10.1016/j.compag.2011.11.011.

[16] N. Iftikhar, T. B. Pedersen, Flexible exchange of farming device data, Computers and Electronics in Agriculture 75 (1) (2011) 52 – 63. doi:http://dx.doi.org/10.1016/j.compag.2010.09.010.

[17] S. Fountas, C. Sorensen, Z. Tsiropoulos, C. Cavalaris, V. Liakos, T. Gemtos, Farm machinery management information system, Computers and Electronics in Agriculture 110 (2015) 131–138. doi:http://dx.doi.org/10.1016/j.compag.2014.11.011.

[18] C.-C. Lin, H. Peng, J. Grizzle, J.-M. Kang, Power management strategy for a parallel hybrid electric truck, Control Systems Technology, IEEE Transactions on 11 (6) (2003) 839–849. doi:10.1109/TCST.2003.815606.

[19] H. Kim, J. Choi, K. Yi, Development of supervisory control strategy for optimized fuel consumption of the compound hybrid excavator, Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering 226 (12) (2012) 1652–1666. arXiv:http://pid.sagepub.com/content/226/12/1652.full.pdf+html, doi:10.1177/0954407012447019.
URL http://pid.sagepub.com/content/226/12/1652.abstract

[20] S. N. Amirkhanian, N. J. Baker, Expert system for equipment selection for earth-moving operations, Journal of Construction Engineering and Management 118 (2) (1992) 318–331. arXiv:http://dx.doi.org/10.1061/(ASCE)0733-9364(1992)118:2(318), doi:10.1061/(ASCE)0733-9364(1992)118:2(318).
URL http://dx.doi.org/10.1061/(ASCE)0733-9364(1992)118:2(318)

[21] M. den Hartog, R. Babuka, H. Deketh, M. A. Grima, P. Verhoef, H. Verbruggen, Knowledge-based fuzzy model for performance prediction of a rock-cutting trencher, International Journal of Approximate Reasoning 16 (1) (1997) 43 – 66, fuzzy Logic Applications. doi:http://dx.doi.org/10.1016/S0888-613X(96)00118-1.

URL http://www.sciencedirect.com/science/article/pii/S0888613X96001181

[22] D. Bradley, D. Seward, The development, control and operation of an autonomous robotic excavator, Journal of Intelligent and Robotic Systems 21 (1) (1998) 73–97. doi:10.1023/A:1007932011161.
URL http://dx.doi.org/10.1023/A%3A1007932011161

[23] P. Kannisto, D. Hästbacka, L. Palmroth, S. Kuikka, Distributed knowledge management architecture and rule based reasoning for mobile machine operator performance assessment, in: Proceedings of the 16th International Conference on Enterprise Information Systems, 2014, pp. 440–449. doi:10.5220/0004870004400449.

[24] P. Kannisto, D. Hästbacka, M. Vilkko, S. Kuikka, Service architecture and interface design for mobile machine parameter optimization system, IFAC-PapersOnLine 48 (3) (2015) 848 – 854. doi:http://dx.doi.org/10.1016/j.ifacol.2015.06.189.
URL http://www.sciencedirect.com/science/article/pii/S2405896315004280

[25] T. Väyrynen, S. Peltokangas, E. Anttila, M. Vilkko, Data-driven approach for analysis of performance indices in mobile work machines, in: DATA ANALYTICS 2015, The Fourth International Conference on Data Analytics, 2015, pp. 81–86.

[26] R. L. Ackoff, From data to wisdom, Journal of applied systems analysis 16 (1989) 3–9.

[27] J. Rowley, The wisdom hierarchy: representations of the DIKW hierarchy, Journal of Information Science 33 (2) (2007) 163–180. arXiv:http://jis.sagepub.com/content/33/2/163.full.pdf+html, doi:10.1177/0165551506070706.
URL http://jis.sagepub.com/content/33/2/163.abstract