AN INTERACTIVE, CLOUD-BASED SIMULATION OPTIMIZATION SYSTEM FOR KNOWLEDGE DISCOVERY AND DECISION SUPPORT IN MANUFACTURING

Ingemar Karlsson[1], Amos H.C. Ng[1], Tehseen Aslam[1] & Catarina Dudas[2]

[1]*Virtual Systems Research Centre, University of Skövde, Sweden*
[2]*Volvo Group Trucks Operations, Sweden*

ingemar.karlsson@his.se

Abstract: Designing or improving a manufacturing system involves a series of complex decisions over time to satisfy the strategic objectives of the company. To select the optimal parameters of the system entities so as to achieve the desired overall performance of the system is a very complex task that has been proven to be difficult, even for a seasoned decision maker. One of the major barriers for more efficient decision making in manufacturing is that whilst there is in principle abundant data from various levels of the factory, these data need to be organized and transferred into knowledge suitable for decision-making support. The integration of decision-making support and knowledge management has been identified to be more and more important in both scientific research and from industrial companies. The concept of deciphering knowledge from multi-objective optimization was first proposed by Deb with the term innovization (innovation via optimization). By integrating the concept of innovization with simulation, a new set of powerful tools for manufacturing systems analysis, in order to support optimal decision making in design and improvement activities, is emerged. This method is so-called Simulation-based Innovization (SBI), which has been proven to produce promising results in our previous application studies. Nevertheless, to promote the wider use of such a new method requires the development of an integrated software toolset. The goal of this paper is therefore to outline a Cloud-computing based system architecture for implementing such a SBI-based Interactive Decision Support System.

Keywords: Decision Support System, Production Systems Simulation, Innovization, Cloud-computing, Data Mining.

1. INTRODUCTION

Designing or improving a manufacturing system involves a series of complex decisions over time to satisfy the strategic objectives of the company. The decisions on, e.g. equipment sizing, layout, level of automation, workload allocations, internal and external material logistics, for a new manufacturing facility or for the re-configuration of an existing one can pose big challenges to the designer/manager because of the complex combinations and interactions among the system entities. Furthermore, to select the optimal parameters of the system entities so as to achieve the desired overall performance of the system is a very complex task that has been proven to be difficult for the decision maker in the design process. As a matter of fact, selecting which decision variables to be included in decision making activities is by itself a complex decision-making problem. This is particularly relevant to manufacturing system design and improvement as the number of system parameters is very often numerous, which making the determination of which parameters are influencing and should be included in the analysis activity too difficult to be answered by existing statistical methods like Design of Experiments (DoE). On the other hand, the fact that many Multi-Criteria Decision Making (MCDM) research efforts only put focus on helping the decision maker to choose among the solutions by performing analysis on the objectives have ignored two important facts: (1) before a decision is made, very often a decision maker wants to know if there are any patterns/rules that relate the decision variables to the Pareto-optimal solutions; (2) the decision space is equally important in decision making. The latter fact means that a user would be likely to select the most suitable values of the decision variables from a set of solutions that are close to each other in the objective space.

One of the major barriers for more efficient decision making in manufacturing is that whilst there is in principle abundant data from various levels of the factory, these data need to be organized and transferred into knowledge suitable for decision-making support. As an example, unraveling or discovering relationships between input parameters and output parameters such as productivity and product quality requirements in manufacturing is seen as an important task. The idea of deciphering knowledge, or knowledge discovery, by the post-optimality analysis of Pareto-optimal solutions from multi-objective optimization was first proposed by Deb and Srinivasan (2006). He coined the term innovization (innovation via optimization) to describe the task of discovering the salient common principles present in the Pareto-optimal solutions so that deeper knowledge/insights on the behavior/nature of the problem can be gained. The innovization task employed in earlier publications involved the manual identification of the important relationships among decision variables and objectives that are common to the obtained trade-off solutions. Recent studies using data mining techniques so that innovization procedures can be performed automatically have been shown to be promising in various engineering design problems (Bandaru and Deb, 2010, 2011).

Data mining has been described as a technology which will trigger a new revolution on how scientific research is conducted. While Microsoft Research calls this “The Fourth Paradigm” (Microsoft Research, 2009), McKinsey & Company uses the term “Big Data” (McKinsey Global Institute, 2011) to describe the big impact of data mining in the future of business and industry. But both of these research reports do not explicitly recognize the power of combining simulation-based optimization (SBO) and data mining. The uniqueness of innovization, as has been shown in several engineering applications, is in using advanced data analysis to decipher salient properties from the optimization data generated, and not data that are already exist in a data source. As a matter of fact, by integrating the concept of innovization with simulation and data mining techniques, the innovization task can be used effectively for the analysis and decision-making support in the system design/development of industrial-scale production or supply-chain systems. To our best knowledge, despite some related case studies, e.g. in health-care services can be found very recently (Lin, et al., 2013), such a post-optimality analysis from data generated from simulation for production systems, has not been attempted in other research.

The research in Skövde aims at pushing the state of the art even further by explicitly integrating interactive decision-making and rule visualization into a general Simulation-based Innovization (SBI) based decision making support system. Particularly, the system will put emphasis on how interactive decision making and rule visualization can fit perfectly into a SBO framework. On one hand, interactive decision making can be incorporated into the SBO and knowledge discovery process. On the other hand, this interactive process can feed back the rules to the original SBO process to help the optimization to converge faster to the preferred area of the decision maker. In terms of scientific progress, this concept has put new unique values to the existing SBO methodology. But to fully realize the power of such an interactive decision making and optimization framework in solving real-world production system problems would require a complete intelligent decision support system (IDSS) for production managers/engineers to use, test and validate in practice. Hence, the aim of this paper is to introduce such a software prototype that is now being developed to demonstrate the above-said concepts.

2. SIMULATION-BASED INNOVIZATION

SBI (Ng, et al., 2009, 2011, 2013) uses SBO as a method for retrieving the optimization data for analysis. The second component in SBI is data mining which extracts knowledge from the results from SBO, as illustrated in Fig. 1.

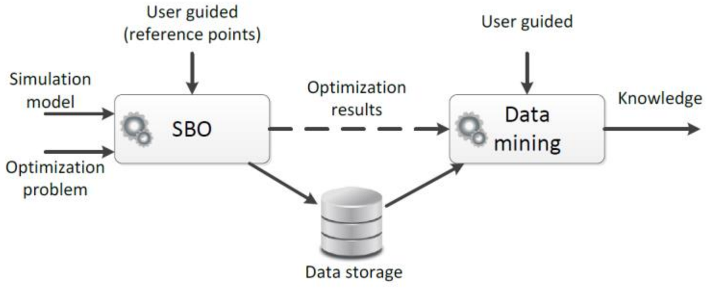


Fig. 1. The SBI Process.

Simulation-based optimization is an iterative process where an optimization algorithm uses a simulation model to evaluate solutions. Inputs are generated by the optimization algorithm, which are fed into a simulation model. The response from the model is then used by the optimization algorithm for generating a new set of inputs. The optimization algorithm might perform many of these iterations to get a solution that meets the decision maker's requirements. The evaluation time depends on the type of simulation and the complexity of the model. If the simulation model is stochastic, each evaluation of a model setup (solution) needs to be replicated so that statistically reliable output data can be generated. Such a resampling also means that the optimization will take even longer time to complete.

A company's use of this kind of software will vary over time. The need increases when the company plans for new products and/or new manufacturing lines. Another common use case scenario will be the continuous performance improvement of an existing manufacturing line. Because of all these different requirements, the utilization of a computer cluster for SBO will also vary. This makes it hard to decide how large the cluster should be. In some situations, the cluster may be too small and at other times it may be unused or with very low utilization. If the cluster is too small, it may hinder the company to make decisions in time and on the other hand an over-dimensioned cluster means the company has to maintain unnecessarily costly computing resources. In the next section, cloud computing is introduced as a possible solution to this problem of computing resource allocation.

3. A CLOUD-BASED, INTERACTIVE PLATFORM

Cloud computing has a broad meaning. In some definitions, both the applications delivered through the cloud and the services and hardware behind it can be referred to as cloud computing (Armbrust, et al., 2010). National Institute of Standards and Technology (NIST) defines cloud computing as "a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction" (Mell and Grance, 2011). Cloud computing with its scalable and elastic features has opened up for new flexible and cost-effective web applications that have not been possible before. The move from physical servers to virtual servers has led to the development of different cloud services. A physical server can host several virtual servers where the exact number is limited by the resources on the host computer. This technique gives organizations the possibility to increase and decrease their computational resources which results in flexibility that is not possible with a traditional data center. Such kind of scalability and elasticity is unique to cloud computing, which opens up for new applications that rely on distributed computation.

By utilizing the concept of cloud computing, the problems with a SBI platform can be mitigated. There are many advantages offered by cloud computing, but the bottom line is, it lets a customer to hire in only the required computational capacity at the required time frame. If the SBO platform is run in a cloud environment, the company does not have to own any cluster hardware and in that way save money, electricity and other resources. Cloud computing usually has a "pay as you go" model (Furht and Escalante, 2010) so that the customer only pays for the services they use. Because of this, it is possible to start only the number of virtual computers that are needed at the required moment, which makes the solution "elastic" but still highly scalable.

In principle, the process of SBI fits well in the distributed cloud environment. In its most basic form, both SBO and data mining can be parallelized and therefore the individual computational work packages can be sent to different computers in a distributed environment. In the following, we propose a new platform that can perform simulation-based optimization and automatically applies data mining techniques on the optimization results. The platform is also designed to make use of the properties of cloud computing, which makes it different from regular distributed computing.

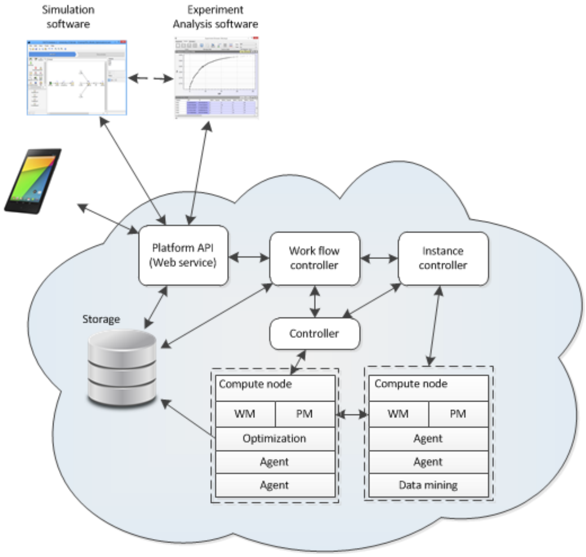


Fig. 2. System architecture of a cloud-based, interactive SBI platform.

The SBI process needs at least two components, SBO and data mining. The SBO process results in a large amount of data from which the data mining process can be used to generate usable knowledge. In Fig. 2, the main parts in the concept platform are shown. This part of the system manages the resources and has two main functions, controlling the computing resources and controlling the optimization and data mining processes. The instance controller launches virtual computer instances based on the computational need from the optimization and data mining processes. These instances are later closed when they are not needed.

The computing resource component assembles all the computational needs and runs them on a virtual computer. There are three functions that need computational resources in the platform: evaluation of simulation models, optimization algorithms and data mining algorithms. The virtual computers that run the computing resource component may run different operating systems, depending on the need for the specific work package. Depending on the cloud environment, there can be several different instances that the platform can choose from. It is the system control component that initiates the launch of the instances. The number of evaluation threads depends on the virtual computer specifications, i.e. number of virtual cores and memory. Both the optimization and data mining algorithms are also controlled by this component. The idea behind the computing resource component is to easily render a virtual computer in different roles, depending on the situation. All the data that are generated in the platform are stored by the data storage component. This can be, but is not limited to, optimization projects with simulation models, optimization results and rules extracted from the data. To be able to make use of the scalability of the cloud environment, the storage should be able to grow together with the amount of data and number of users as well. By storing the data different users can view and modify the same data and because of this collaborate on the same project. The data storage component can use different technologies for storing the data like NoSQL and relational databases, as well as cloud storage facilities like Amazon S3 (Simple Storage Service). New technologies like NoSQL databases have emerged to handle large data (Nambiar, et al., 2014), often called big data. This kind of technologies can handle elastic scaling which is difficult with a traditional storage solution. Therefore the NoSQL databases fit well into the scalable and flexible environment that cloud computing relies on (Pokorny, 2013).

The platform can be interfaced by many types of clients through an API in the system control component. To be able to use the platform, the user needs a simulation model that is supported by the platform. The design of the platform supports all simulation software that are interfacable with an API that supports starting simulations, changing setting, as well as retrieving inputs and outputs. The optimization can be initiated directly from the supported simulation software which sends the simulation model together with the optimization problem (e.g. objectives, the selected algorithm and its parameter setting) to the platform.

The experiment analysis software is used to view the results from the optimization as well as performing analysis on the results. This software can also initialize data mining analysis and view the resulting rules extracted by the data mining process. The extracted rules are stored in the database linked to the experiment.
The platform should be able to let the decision maker guide the optimization during its progress. For example, by supplying one or more reference points to a genetic algorithm, it is possible to find better solutions in less time

(Siegmund, et al., 2012). This can improve the performance of the platform as well as letting the decision maker to point out the preference regions in the objective space to guide the local search. The reference points can be modified throughout the optimization to make sure that the decision maker can provide his/her knowledge during the optimization.

4. SOFTWARE PROTOTYPE & EXAMPLES

System development as a valid research methodology has been well-discussed by researchers in Management Information Systems (Nunamaker, et al., 1991). As they concluded, "*Building a system in and of itself does not constitute research. The synthesis and expression of new technologies and new concepts in a tangible product, however, can act as both the fulfillment of the contributing basic research and as an impetus to continuing research*". This view fits comfortably with the goal of the current project in which a system has to be built in order to test, measure, evaluate and validate the research ideas. System development could be thought of as a "proof-by-demonstration" and systems prototyping has been argued to be a superior methodology in IDSS research (Ekbia, 2006). Because the quality of the support we can provide to managers depends on our understanding of both "decision-making" and "system building", in the development of a software prototype, the integration of "decision-making" and "system engineering" aspects in a unified framework is emphasized. Therefore, for the development of a software prototype, a modified version of the decision support engineering approach proposed in (Gachet and Haettenshwiler, 2006) is adopted as illustrated in Fig. 3. In such a "tripartite approach", the IDSS kernel provides the support analysis to gain knowledge about users' expectations and clearly defines the operations that will be carried out by IDSS.

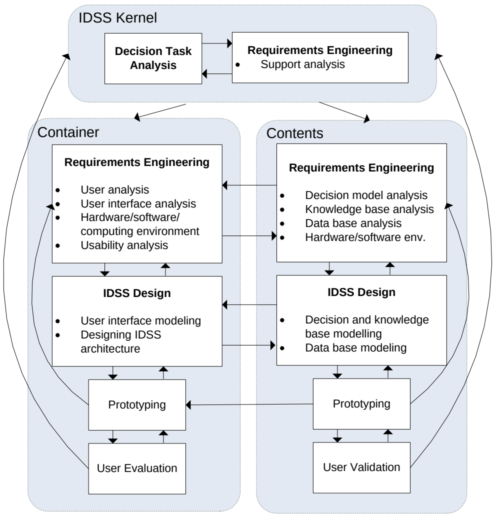


Fig 3. A decision support engineering framework, modified from [18].

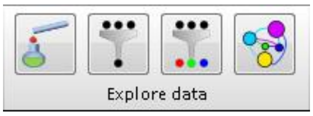


Fig.4. Icons of data analysis and mining function in the software prototype.

Fig. 4 shows the icons of the data analysis and data mining functions which are now being implemented in the prototype of "container" within IDSS, which provide the data visualization and data mining functions for generating the results in some industrial application examples.

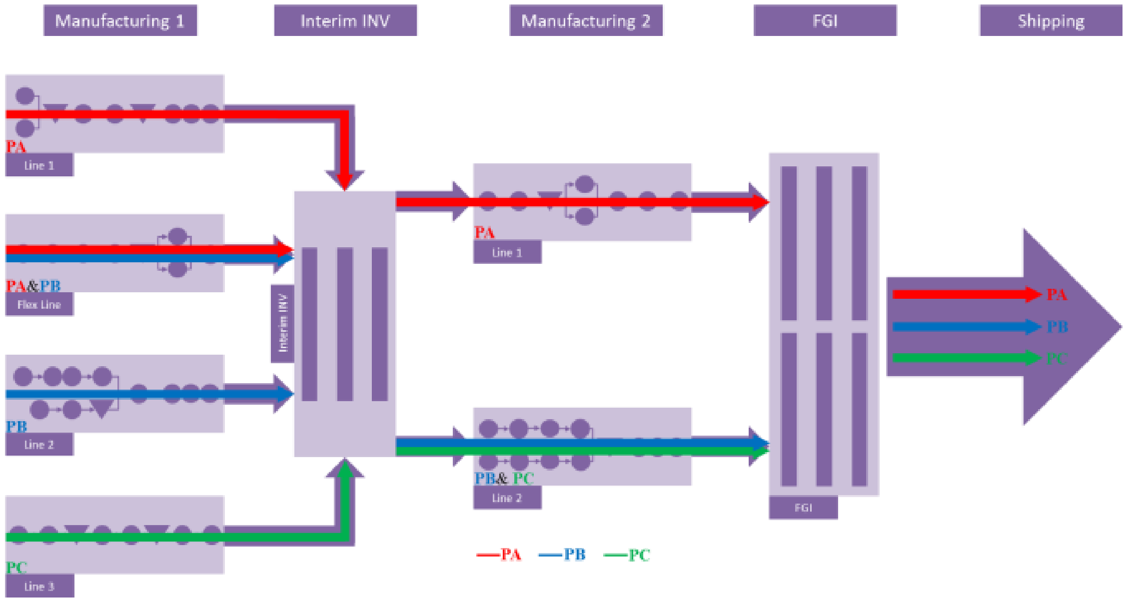


Fig. 5. Process flow of a real-world industrial study.

The first investigated system, which the process flow is depicted in Fig. 5, contains two manufacturing areas, divided by an interim inventory, a finished goods inventory (FGI), as well as a shipping area. The material or product flow of the system is initiated in the first manufacturing area, Manufacturing 1, and continues through the Interim Inventory to the second manufacturing area, Manufacturing 2, which in turn supplies the FGI with finished products before they are shipped off to customers. Manufacturing 1 consists of four production lines; Line 1, Line 2, Line 3 and Flex-line, where lines 1, 2, and 3 are product-specific, i.e. only a single product variant is processed on these lines, whereas the flex-line has the possibility to accommodate and process multiple products. Similarly, Line 1 and Line 2 in the second manufacturing area have a comparable allotment of production in which line 1 is product-specific and line 2 processes multiple products. Each production line has its own internal production set-up and flow. A system dynamics model has been developed for this industrial application study (see Fig. 6) based on the stock management model for inventory and production control introduced by Sterman (2000).

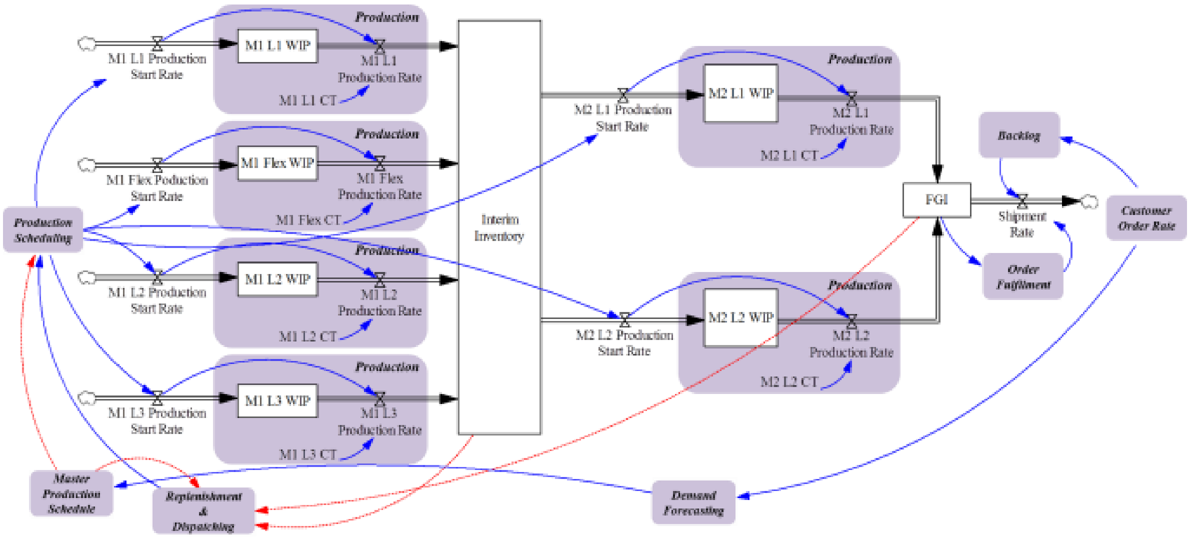


Fig. 6. Structure diagram of the System Dynamics model developed for the studied industrial system.

The aim of this study was to identify the so-called optimal WIP zone (Pound and Spearman 2007). According to Factory Physics (Hopp and Spearman, 2000), for any supply chains, or any other flow processes, the relationship between lead time and the throughput can be determined by the WIP levels in the system. As illustrated in Fig. 7, in any WIP versus Throughput plots, three WIP zones can be divided, namely, WIP starvation zone, the WIP overload zone and the WIP optimal zone. The WIP starvation zone is characterized by the fact that lead time is reduced to a minimum due to low levels of WIP, which subsequently drastically reduces the TH of the system. The WIP overload zone is characterized by the fact that an increase in WIP levels will only increase the lead time and this increase in WIP will have very little or no effect on the system TH. However, in the WIP optimal zone, the system WIP generates the optimal performance of the system by acquiring minimal lead time and maximum system TH (Pound and Spearman 2007).

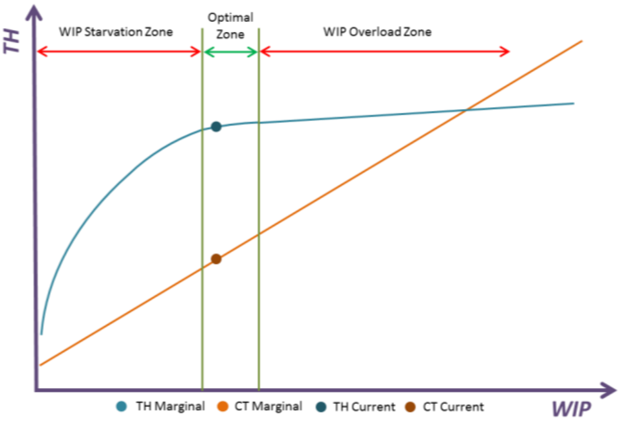
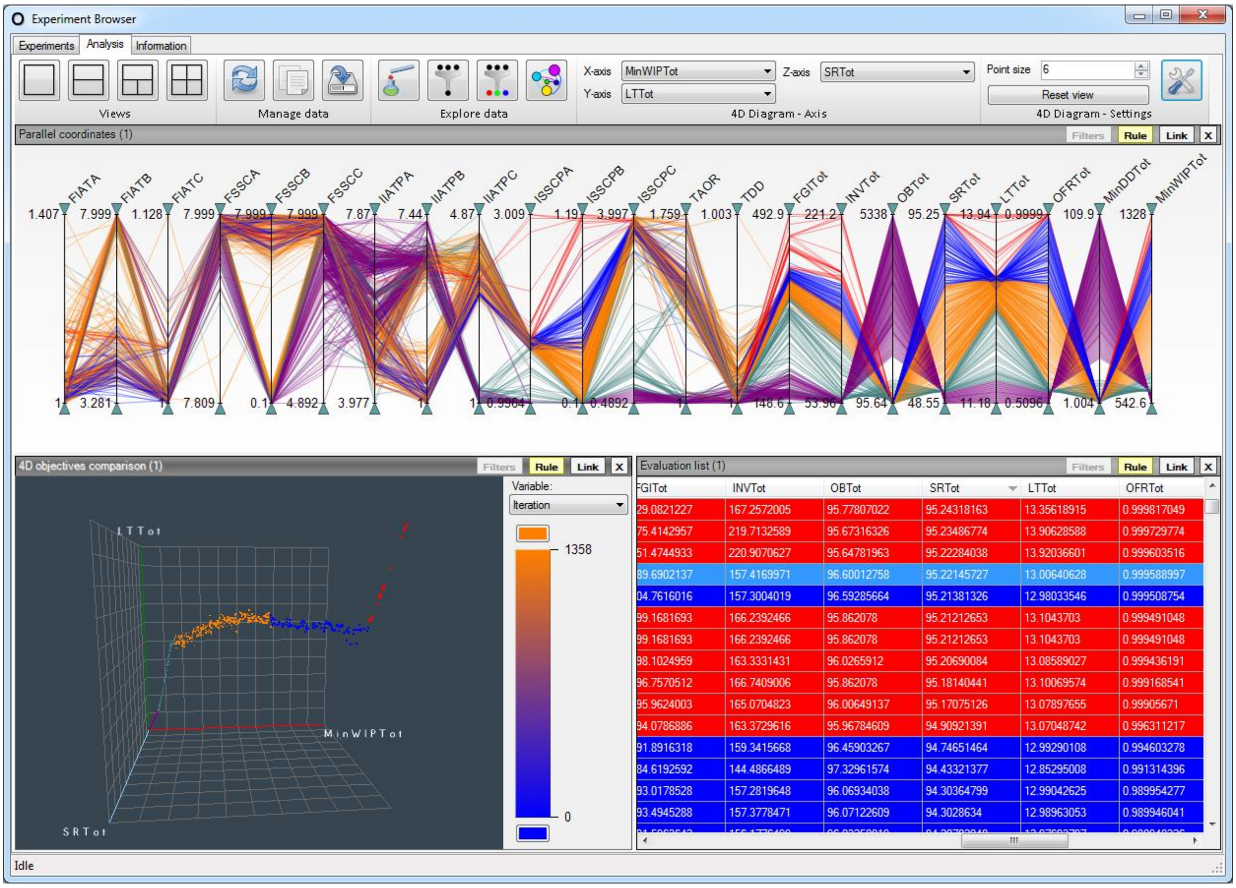


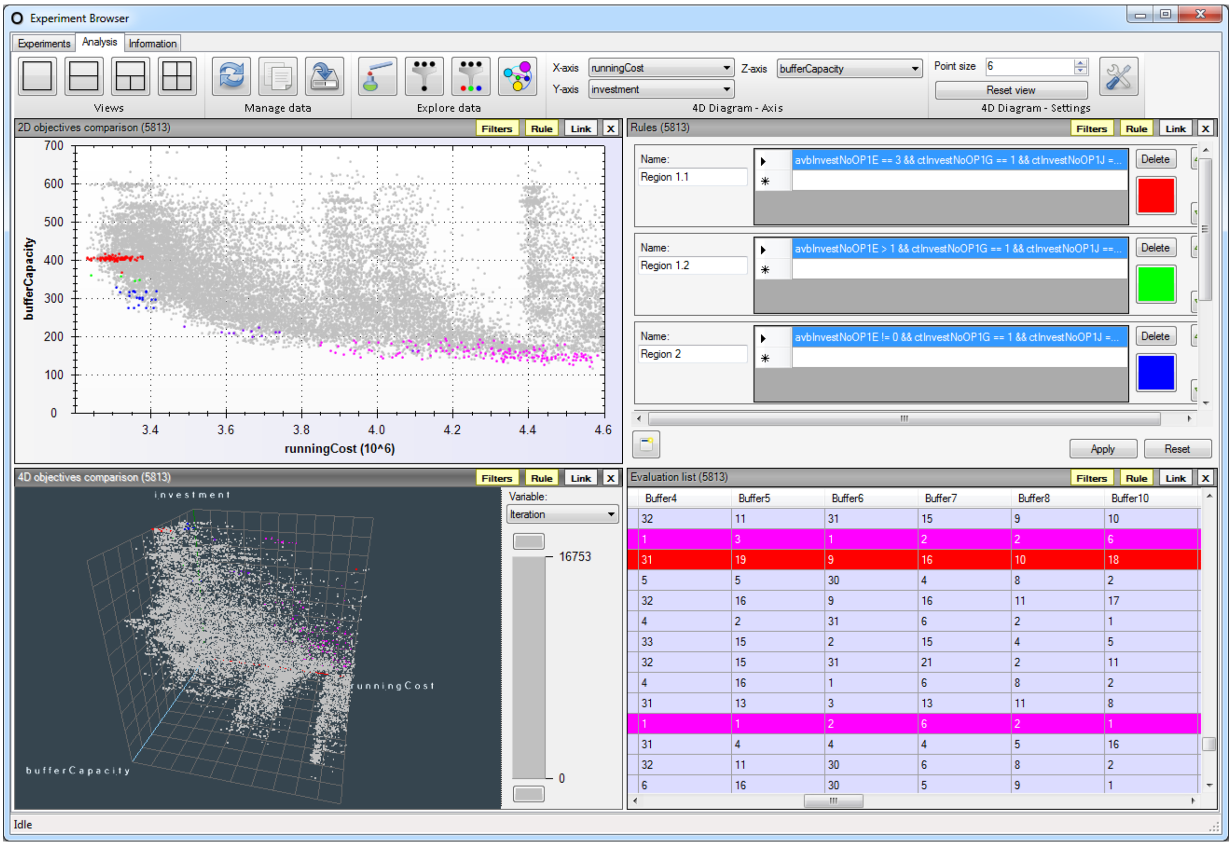
Fig. 7. WIP regions, adopted from (Pound and Spearman, 2007).



X-axis MinWIPTot
Y-axis LTTot
Z-axis SRTot
Point size 6
Variable: Iteration

FGITot	INVTot	OBTot	SRTot	LTTot	OFRTot
29.0821227	167.2572005	95.77807022	95.24318163	13.35618915	0.999817049
75.4142957	219.7132589	95.67316326	95.23486774	13.90628588	0.999729774
51.4744933	220.9070627	95.64781963	95.22284038	13.92036601	0.999603516
89.6902137	157.4169971	96.60012758	95.22145727	13.00640628	0.999588997
04.7616016	157.3004019	96.59285664	95.21381326	12.98033546	0.999508754
99.1681693	166.2392466	95.862078	95.21212653	13.1043703	0.999491048
99.1681693	166.2392466	95.862078	95.21212653	13.1043703	0.999491048
98.1024959	163.3331431	96.0265912	95.20690084	13.08589027	0.999436191
96.7570512	166.7409006	95.862078	95.18140441	13.10069574	0.999168541
95.9624003	165.0704823	96.00649137	95.17075126	13.07897655	0.99905671
94.0786886	163.3729616	95.96784609	94.90921391	13.07048742	0.996311217
91.8916318	159.3415668	96.45903267	94.74651464	12.99290108	0.994603278
84.6192592	144.4866489	97.32961574	94.43321377	12.85295008	0.991314396
93.0178528	157.2819648	96.06934038	94.30364799	12.99042625	0.989954277
93.4945288	157.3778471	96.07122609	94.3028634	12.98963053	0.989946041

Fig. 8. Parallel coordinate and 4D plot visualization for studying optimal WIP zone.



X-axis runningCost

Y-axis investment

Z-axis bufferCapacity

Point size 6

Variable: Iteration

Name: Region 1.1
avbInvestNoOP1E == 3 && ctInvestNoOP1G == 1 && ctInvestNoOP1J =...

Name: Region 1.2
avbInvestNoOP1E > 1 && ctInvestNoOP1G == 1 && ctInvestNoOP1J ==...

Name: Region 2
avbInvestNoOP1E != 0 && ctInvestNoOP1G == 1 && ctInvestNoOP1J =...

Buffer4	Buffer5	Buffer6	Buffer7	Buffer8	Buffer10
32	11	31	15	9	10
1	3	1	2	2	6
31	19	9	16	10	18
5	5	30	4	8	2
32	16	9	16	11	17
4	2	31	6	2	1
33	15	2	15	4	5
32	15	31	21	2	11
4	16	1	6	8	2
31	13	3	13	11	8
1	1	2	6	2	1
31	4	4	4	5	16
32	11	30	6	8	2
6	16	30	5	9	1

Fig. 9. Visualization of rules, generated from decision-tree based data mining, in 4D plot.

By applying the IDSS prototype to the industrial case study, five critical WIP zones where identified as shown in Fig. 8. These 5 zones governed the system behaviour if one checks how the variation of the $Min\ \mu_{TotsWIP}$ parameter can affect the other objectives: WIP Zone 1 [543-591.19], WIP Zone 2 [>591.19-675.44], WIP Zone 3 [>675.44-1015], WIP Zone 4 [>1015-1239] and WIP Zone 5 [>1239-1328]. As demonstrated in the upper plot in Fig. 8, these WIP zones are highlighted with different colours in the Parallel Coordinate visualization, together with their corresponding output and design variable, whereas the lower-left plot displays as 3D scatter plot of $Min\ \mu_{TotsWIP}$, μ_{TotLT} and μ_{TotsR} which represents, respectively, the WIP, the lead time, and the TH in this case study. Now, an examination of the two figures reveals that WIP Zone 1 is characterized by minimal total lead time for a low total system WIP, but also a very low total shipment rate, whereas the Pareto-optimal solutions in WIP Zone 2 obtain drastically increased lead time for a small increase in total system WIP and with almost no increase in total shipment rate. The later behaviour, i.e., of WIP Zone 2, is more evident in the 4D (3D plus colour) in the lower-left sub-plot of Fig. 9, where its shows how the gradient of WIP Zone 2 is almost horizontal, indicating a very small increase in the total shipment rate while showing a radical increase in total lead time. Similarly, one sees that the gradient of WIP Zone 2 slightly increases for the total system WIP, but for a slight increase in this variable one sees that the total lead time increases drastically.

While the first example shows how data visualization on MOO data can be very useful for supporting decision making. The second application example explicitly uses data mining, in this case, a decision-tree based SBI method, described in (Ng, et al., 2012). The details of the industrial decision support problem have been provided in (Ng, et al., 2012) so they are not repeated in the current paper. The snapshot in Fig. 8 illustrates how the developed IDSS software prototype is used to highlight the solution clusters in the 4D plot (lower-left sub-plot), representing the rules generated from the SBI method and displayed in the upper-right plot. As described in (Ng, et al., 2012), the rule set (Region 1.2), highlighted as green solutions in the 2D and 4D plots, were the preferred solutions that the production manager made the decision in selecting which improvement combinations had to be implemented to achieve the desired objectives. In this way, the current research has made the previously published SBI procedure in an automated way to further support real-world manufacturing decision-making activities.

5. SUMMARY

By integrating the concept of Innovization with simulation, a new set of powerful tools has emerged for the design, analysis, optimization and improvement of production systems. As a method for retrieving the relationships between decision variables and system objectives, such as any performance measures of production systems, SBI can be regarded as a tool for supporting real-world decision making. This concept has been introduced and demonstrated in our previous SPS publications over the years. In this paper, we have outlined our plan in how a software prototype can be developed to automate the SBI procedure, developed on top of an interactive, cloud-based computing platform. As can be seen from the paper, the implementation that allow rules to be generated from data mining to visualization, in form of high-dimensional data plots, has been partially completed. Current work in the user interface is focused on, e.g., mapping data visualization to production model visualization. As the target is an Intelligent DSS system, more research work has to be done on "contents" side, regarding how rules/patterns/knowledge generated can be stored, searched and retrieved for later use.

6. ACKNOWLEDGEMENT

This work was initiated during the FFI-HSO project (2009-2012) funded by VINNOVA. The research is continued within the IDSS project funded by KKS. The authors gratefully acknowledge these financial supports.

REFERENCES

Armbrust, M., Fox, A., Griffith, R., Joseph, A. D., Katz, R., Konwinski, A., Lee, G., Patterson, D., Rabkin, A., Stoica, I., and Zaharia, M. (2010). A view of cloud computing. *Commun. ACM*, **53**(4):50–58.

Bandaru, S. and Deb, K., (2010). Automated Discovery of Vital Knowledge from Pareto optimal Solutions: First Results from Engineering Design. In *IEEE Congress on Evolutionary Computation*, CEC '10, 1-8.

Bandaru, S. and Deb, K., (2011). Towards automating the discovery of certain innovative design principles through a clustering based optimization technique, *Engineering Optimization* **43**(9), 911-941, 2011.

Deb, K., and Srinivasan, A., (2006). Innovization: Innovating Design Principles through Optimization. In *Proceedings of the 8th annual conference on genetic and evolutionary computation*, Seattle, USA, July 2006, 1629-1636.

Ekbia, H. (2006). Taking Decisions into the Wild: An AI Perspective in the Design of i-DMSS. *Intelligent Decision-making Support Systems: Foundations, Applications and Challenges*, J.N.D. Gupta, G.A. Forgionne and M. Mora (eds.), Springer, 79-96.

Furht, B. and Escalante, A. (2010). *Handbook of Cloud Computing*. Springer Publishing Company, Incorporated, 1st edition.

Gachet, A. and Haettenshwiler, P. (2006). Development Processes of Intelligent Decision-making Support Systems. *Intelligent Decision-making Support Systems: Foundations, Applications and Challenges*, J.N.D. Gupta, G.A. Forgionne and M. Mora (eds.), Springer, 97-121.

Hopp, W.J. and Spearman, M.L., (2000). *Factory physics: foundations of manufacturing management*, 2nd ed., Irwin McGraw-Hill Higher Education, Burr Ridge, IL.

Lin, R.-C., Mustafa Y.S. and Pasupathy, K.S. (2013). Multi-objective simulation optimization using data envelopment analysis and genetic algorithm: Specific application to determining optimal resource levels in surgical services, *Omega*, **41**, 881-892.

McKinsey Global Institute (2011). *Big Data: the next frontier for innovation, competition, and productivity*, McKinsey & Company, www.mckinsey.com/mgi.

Mell, P. and Grance, T. (2011). *The NIST definition of cloud computing—recommendations of the national institute of standards and technology*. Special publication 800-145, NIST, gaithersburg.csrc.nist.gov/publications/nistpubs/800-145/SP800-145.pdf. [accessed 2014-05-15].

Microsoft Research (2009). The Fourth Paradigm: Data-Intensive Scientific Discovery, *Microsoft Research* (2009), Redmond, Washington. Tony Hey, Stewart Tansley and Krist in Tolle (eds.). http://research.microsoft.com/en-us/collaboration/fourthparadigm/

Nambiar, R., Chitor, R. and Joshi, A. (2014). Data management – a look back and a look ahead. In Rabl, T., Poess, M., Baru, C., and Jacobsen, H.-A., editors, *Specifying Big Data Benchmarks*, No. 8163 in Lecture Notes in Computer Science, Springer Berlin Heidelberg, 11–19.

Ng, A.H.C., Deb, K. and Dudas, C. (2009). Simulation-based innovization for production systems improvement: An industrial case study, in *The International 3rd Swedish Production Symposium*, The Swedish Production Academy, 278–286.

Ng, A.H.C., Dudas C. and Deb, K. (2011). Simulation-Based Innovization using Data Mining for Production Systems Analysis. In *Evolutionary Multi-objective Optimization in Product Design and Manufacturing*. L. Wang, A. Ng, K. Deb (eds), Springer, 401-430.

Ng, A.H.C., Dudas, C., Boström, H. and Deb, K. (2013). Interleaving Innovization and Multi-Objective Decision-Making for Faster Convergence in Production Systems Optimization. In *Learning and Intelligent Optimization,* G. Nicosia and P. Pardalos (Eds): LION7, Lecture Notes in Computer Science, **7997**, 1-18.

Ng, A.H.C., Dudas, C., Pehrsson, L. and Deb, K. (2012). Knowledge Discovery in Production Simulation by Interleaving Multi-Objective Optimization and Data Mining. In *Proceedings of the 5'th Swedish Production Symposium (SPS'12)*, Linköping, Sweden, 6-8 November 2012, 461-471.

Nunamaker, J.F., Chen, M., and Purdin, T.D.M. (1991). System development in information systems research. *Journal of Management Information Systems*, **7**(3), 89-106.

Pokorny, J. (2013). NoSQL databases: a step to database scalability in web environment. *International Journal of Web Information Systems*, **9**(1), 69–82.

Pound, E.S. and Spearman, M.L., (2007). *The Great Push vs. Pull Diversion*. Factory Physics, Inc. http://www.factoryphysics.com/ [accessed 2013-12-05]

Siegmund, F., Bernedixen, J., Pehrsson, L., Ng, A. H.C. & Deb, K. (2012). Reference point-based evolutionary multi-objective optimization for industrial systems simulation. In *Proceedings of the 2012 Winter Simulation Conference*.

Sterman, J., (2000). *Business Dynamics: Systems Thinking and Modeling for a Complex World*. Irwin/McGraw-Hill.