The Construction of a Common Objective Function for Analytical Infrastructures

Marcus Grum, Benedict Bender
Department of Business Informatics,
Especially Processes and Systems
Potsdam, Germany
mgrum@lswi.de, bbender@lswi.de

Attahiru Alfa
Department of Electrical, Electronic and Computer Engineering
University of Pretoria
Pretoria, Gauteng, South Africa
attahiru.alfa@up.ac.za
and
Department of Electrical and Computer Engineering
University of Manitoba,
Winnipeg, Manitoba, Canada
attahiru.alfa@umanitoba.ca

***Abstract*—The paper deals with the increasing growth of embedded systems and their role within structures similar to the Internet (Internet of Things) as those that provide calculating power and are more or less appropriate for analytical tasks. Faced with the example of a cyber-physical manufacturing system, a common objective function is developed with the intention to measure efficient task processing within analytical infrastructures. A first validation is realized on base of an expert panel.**

***Keywords*—Analytic Infrastructures; Cyber-Physical Manufacturing Systems; Measuring Efficient Task Processing**

I. INTRODUCTION

An analytical system converts analytical tasks to finished analytical results. Since modern production systems integrate independent, cooperating cyber-physical systems and each is more or less appropriate to carry out analytical tasks next to the conversion of raw materials, the analytical tasks of a production system would either be in value added or in non-value added state, while states differ in ascribed values. This research focuses on reducing non-value adding states in multi-analytic and complex production systems and tries to address the research question:

Which common objective function shall be used to optimize task processing within analytical infrastructures?

This includes the reduction of traditional bottlenecks, since blocking or starving machines increase non-value added time within the entire system, but considers bottlenecks with respect to the flexibility of analytical task infrastructures and shall increase the system performance as well as measure task distribution approaches such as [1].

Since this research approach is intended to be design-oriented as Peffers proposes [2] and [3], the paper is structured as follows: A second section presents underlying concepts, the third sections provides the creation of a common objective function and its demonstration. A final chapter concludes the paper.

II. RELATION TO EXISTING THEORIES AND WORK

The following builds on theory of analytical infrastructures, and lists characteristics of analytical tasks. Further, approaches of efficient task processing are discussed from the perspective of analytical infrastructures. Existing measurement approaches shall serve as starting point for the creation of a common objective function.

A. Analytical Infrastructures

In order to decide where tasks should be processed and how the prioritization should be conducted it is necessary to understand the underlying IT infrastructures. Even though the individual company's infrastructure may vary, common patterns and levels of processing infrastructure should be used as a basis for the objective function and the related characteristics. As a basis for the development of the common objective function, three levels of computing infrastructure are assumed.

The lowest level subsumes different machines and components, that more or less directly participate in the, typically physical, value creation process, as it was described by [4] components at this level typically show various elements of cyber-physical systems (CPS). Those CPS are able to sense their environment, communicate with other CPS, react to the environment, and locally process information or take a decision [5]. The combination of various CPS that commonly participate in the value creation process is also referred to as cyber-physical production system (CPPS), which here serves as application context.

A further level, called local cloud, subsumes more centrally located processing infrastructure components in a company. Those resources are typically more powerful than single CPS and employ central company systems. This includes mostly data warehouse und business intelligence or reporting software.

A third level itself is not part of the company's infrastructure. Computing infrastructures on the public cloud level are typically rented from cloud hosting provider that offer computing resources on demand and are billed by a pay-peruse model. In general, those infrastructures show high scalability. This allows them to be used when needed but the user is not urged to pay for them when not.

The processing of analytical tasks can be carried out on any of those three levels taking into account the platform's individual *resource characteristics* as well as their *resource constraints*, such as their current work load, acquisition costs, etc. Typically, the computing power rises with increasing levels, as Fig. 1 shall visualize. With this, further hardware costs and transfer times go along, which are based on *network infrastructure* components as their internet connections, communication components, physical distance, etc. This implies that a common objective function should consider those hardware requirements of analytical infrastructures.

Since those infrastructure characteristics influence the evaluation of an efficient task processing, each serves as base for the creation of a common objective function.

B. Task Characteristics

Analytical tasks can be classified according to different sets of criteria. For the purpose of this paper, characteristics, which influence the optimal point of processing, are especially relevant. The following aspects are useful in order to describe tasks and their related properties as a pre-step in order to find optimal or at least well-suited processing points.

One important criterion is the *remaining time* until a missing result would influence other components in the processing infrastructure. From this point in time, a missing result would have a negative impact on the whole system and it should therefore be ensured that the result is available at latest at this point in time. This can be seen as a due date (time).

The importance of a result is related to the first aspect. While it might be less relevant whether a report is created at midnight or a few minutes later, it might be highly relevant if a decision within an active production setting is missing and the production is getting stuck. This task characteristic could be described as *importance* for the given scenario or the system as a whole.

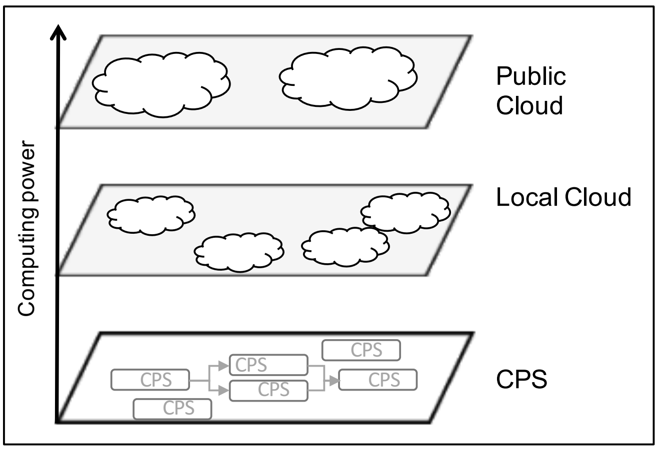

Fig. 1. Model of analytical infrastructures (in accordance to [1]).

Computing effort is an important criterion in order to decide where to process tasks. For complex, high demanding tasks other infrastructures are suitable then for less demanding, simple decision tasks. Especially in the context of CPPS this is relevant since lots of CPS are involved in the process that provide very limited resources. Fig. 1 shows the three level and the corresponding assumed computing power. Besides the aspect of computing power, the conducted task influences the computing power needed. Simple tasks like averaging several values might require less power than complex statistical analyses.

The *amount of data* involved in the task processing is also relevant for task processing because of the following assumption: The more data is involved in the task processing, the more power is needed to compute the required result.

Further, a transmission might be necessary between the data of the data creation point and the processing point. Here, the transmission of data from the origin to the processing point, the processing itself as well as the transmission of the result, as the processing point was not the final destination point, was required. Since networking infrastructures vary in their performance, the aspect should be considered.

Another aspect within network infrastructures with a huge amount of nodes is the distribution of processing tasks over multiple entities (e.g. map reduce-principles). Therefore it is important whether a task could be well *parallelized* or not. Huge tasks could be processed faster if they are distributed over multiple computing components.

Finally, from the *task derived system requirements* should be taken into account. A task processing might require special software components and execution environments as well as frameworks in order to process tasks. Depending on the availability of those circumstances, the abstraction level that could be used in order to conduct the task processing might vary widely. The system requirements should therefore also be taken into account when deciding where to process a task.

Since those task characteristics influence the evaluation of an efficient task processing, each can have a fundamental contribution to optimizations based on a common objective function.

C. Efficient Task Processing

Being confronted with a great number of approaches to process tasks efficiently, approaches focus various processing areas, show different complexities and differ in their range. While every day processing approaches are mostly simple and focus the context of only one person, such as the 1-3-5 daily to-do task lists or the idea to realize tasks once as they need no more than three minutes, Big Data approaches focus the even more complex interplay of several systems, but are not as complex as efficient task processing approaches of the real-world human brain.

Within Analytical Infrastructures, the following focuses on Big Data approaches, which show empirically proven good results. It can be interpreted as efficient task processing algorithms: uniform memory approach (UMA), distributed memory approach (DMA) and map reduce approach (MRA).

UMA refers to multi processor systems that realize access of each processor on one common, global long-term memory [7]. As bandwidth and latency are in harmony, bottlenecks are avoided, those systems are easy scalable and can perform without further data transfers.

Multi processor systems, that provide separate long-term memories for each performing unit, refer to DMA. Here, a further data transfer to the local memories is required before computation. As data is spread wisely, bottlenecks of UMA can be avoided efficiently [7].

As long lasting, numerous data transfers in DMA are required, or local memory capacities are exceeded, MRA are attractive. As separate map and reduce steps as well as a complex role and process models are required for this (see [8], p. 65), various implementation requirements are needed as EMC Education Service lists ([6], p. 300):

- Job allocation w.r.t. the current workload
- Job monitoring
- Failure robustness
- Possibility for corrections
- Decentral data memory
- Decentral data processing
- Preference for short ways
- Interim and final result provision

Selected, such as non-selected, efficient task processing algorithms can imply for a common objective function the following: Their basic ideas can serve to derive objectives for a common objective function. Further those can serve as application objects that historically already have been optimized by great innovators and in consequence should show kinds of optimal results measured by a common objective function. Since each processing approach focuses on the range of an entire system, the common objective function may not be limited by the CPS individual perspective and must focus the entire system as well.

D. *Existing Measurement Approaches*

As attractive bottlenecks shall be identified within analytical infrastructures, traditional approaches focus the following: The longest queue method identifies machines with the longest queue as bottleneck [9], while [10] identify bottlenecks at machines with the lowest production or throughput rate within the system. The lowest blocking and starving time [11] or the highest utilization of a machine [12] can serve as indicator for bottlenecks as well. [13] identify machines with the longest active duration as bottleneck. Alternatively [14] propose the inactive duration method based on a bottleneck time ratio, bottleneck ratio, bottleneck shifting frequency and bottleneck severity ratio to identify bottlenecks. All of those approaches can be focused on the single process step at a system as well as on the process chain of the entire system.

Throughput and **Process Rate**: The average number or processed analytical tasks relative to the required time.

Utilization: The running time of a server relative to the available time.

Starving Time: The cumulated time of analytical tasks, because of a lack of material (e.g. data, resources).

Blocking Time: The cumulated time of analytical tasks, which cannot be processed because of missing storage place.

Queue Length and **Waiting Time**: The mean number of waiting jobs and the mean waiting times of jobs.

Average Work In Process and **Active Duration**: the average number of jobs in a system and the average time of jobs.

Following [15] and [16], the allocation of buffer and adding further capacity can mitigate bottlenecks efficiently. Because of a non-optimal use of those as bottleneck mitigate strategies, inefficient task processing approaches can result as well.

Capacity: The possible number of analytical tasks relative to a period of time.

Buffer: The number of resources, tasks or further inventory between two stages.

Each KPI can be combined with an individual objective. For example for some stands the objective to gain high throughput rates because systems perform cheaply w.r.t. fix costs, while for others throughput rates of expensive systems are intended to be avoided. Hence, each KPI can be interpreted as proper *optimization dimension* with *correlations* to others. While KPI do not necessarily show the same *entities*, this further implies for a common objective function to be able to deal with different entities.

III. Research/ Technology/ innovation approach

Following the design-oriented approach ([2] and [3]), the following shows the establishment of objectives for a common objective function for analytical infrastructures and its design. As demonstration, an expert panel has been carried out to validate influencing factors and the objective function itself. An evaluation considers finding w.r.t. the research question.

A. *Establishment of Objectives*

A collection of requirements for a common objective function for analytical infrastructures has been created and can be seen in the following. The foundation of each can be found within the second subsection.

1. The objective function has to consider hardware requirements of analytical infrastructures, such as *resource characteristics,* as well as their component's *resource* constraints and underlying *network infrastructure*.
2. The objective function should consider the point in time when the result is needed (*remaining time*). Using this information, different objectives could be targeted. For example a task could be processed as early as possible in order to use free resources. Alternatively, it could be

processed as late as possible in order to follow the just in time paradigm.

3. The objective function has to consider the *tasks importance* (criticality). Tasks that have a more serve result if not conducted should be prioritized higher than tasks that would have little influence on the system's whole target.

4. *Costs* are also an important decision criterion in modern computing infrastructures. Therefore the goal of cost minimal computation could also be an aspect for the objective function.

5. Since the possibility to react on changing conditions is an important possibility of CPPS infrastructures, the environment needs to be responsive at any time. Therefore the computing components should not be *overloaded*.

6. The objective function has to consider implications based on a *data transmission*. The advantages of a transmission, computing and backwards transmission should outweigh the additional transmission requirements.

7. The objective function has to deal with unnecessary transmission steps.

8. If the processing can be parallelized, the amount of systems that are participating should be as low as possible, but as great as necessary in order to satisfy other criteria such as cost-minimal, time optimal, etc.

9. The objective function has to consider the entire system of efficient task processing units, such that global optima can be identified and are not confused by local optima.

10. The objective function has to consider the meaning of a working unit within the Big Data Approaches (Uniform Memory, Distributed Memory and Map Reduce Approaches).

11. Since optimizing an objective function means optimizing existing KPIs, which includes bottleneck approaches, those can be considered within the objective function, while considering *contradictory* objectives, different *entities* and *correlations* of objectives.

As a designed objective function is intended to be validated, the following criteria serve for the design of an expert panel:

12. The experts shall bring in their expertise without being influenced of current approaches.

13. The experts shall bring in their feedback about the base of the objective function.

14. The experts shall bring in their feedback about the current objective function.

B. Design of a Common Objective Function

Based on the establishment of objectives for a common objection function, the following section designs it as equation.

Faced with various approaches to identify bottlenecks, the current equation builds on the use of the KPI "Waiting Time" *w* (objective 11), although others would have been possible as well. Since tasks can be realized in a parallel, the objective function works on base of job parts (objective 8) and considers *w* on a job part level of every CPS within the system (multiple parts build a whole job). This can be seen in (1) at the use of the sum of job part *j* over all job parts m_i.

As objective 9 asks for, the measurement of the efficient task processing is not realized for every CPS separately, but for the whole system. Hence, the objective function sums over the entire analytical system with *n* CPSs and builds up on the processing of every CPS *i*. Hence, the assessment can consider the role fulfillment of any CPS as well (objective 10). Since the identification of global optima needs to deal with temporal useful high job loads, which are cheaper than more expensive systems, the objective function considers a period of time. Hence, the assessment is realized on time step *t* and considers in total *o* time steps:

$$f(w) = \sum_{t=1}^{o} \sum_{i=1}^{n} \sum_{j=1}^{m_i} a_{i,j}^t w_{i,j}^t, \qquad (1)$$

where

$$a_{i,j}^t = \frac{I_j^t C_{i,j}^t}{R_j^t \; min(T_{i,j,k}^t)}, \qquad (2)$$

and at *time t*, R_j^t is the *remaining time* of job part *j*, $C_{i,j}^t$ is the *processing cost* of job part *j* at CPS *i*, I_j^t is the *importance* of job part *j* and $T_{i,j,k}^t$ is the *transfer cost* of job part *j* from CPS *i* to CPS *k*. The assessment of efficient task processing is realized for any CPS *i* for any job part *j* at time step *t* and can be interpreted as follows:

This *waiting time* is captured by the corresponding *remaining time*. A great remaining time indicates flexibility and hence, job parts with less flexibility can efficiently be preferred. As the *importance* of this job part is great, for example because it is part of the critical path, it is efficiently to be preferred. The *processing cost* of a job part is in relation to the minimal *transferring costs* from the current CPS to other CPS *k*. The smaller the relation is, the more efficient is the processing at the current CPS.

The interest is to minimize *f(w)*, hence the intended objective function would be

$$min f(w) \qquad (3)$$

Hence, the objective function can be used for the measurement as well as the optimization of efficient task processing analytical infrastructures.

C. Design of an Expert Panel

With the intention to validate the designed objective function before its implementation and to consider further experiences, a three-step expert panel is designed, as can be seen in Fig. 2. While experts of analytical infrastructures are to be faced with tasks step by step here, the authors of the paper are to moderate sessions, explain approaches and are intended to deal with questions so that a common understanding is realized. Each step focuses on an objective for the validation of a common objective function with an expert panel. The experts are to be faced with three steps:

A first step is designed to identify expert knowledge. As kind of introduction question, the experts are asked to begin with an identification of indicators or influencing factors of efficiently working analytical infrastructures. Then, those have to be prioritized in allocating points from 1 (not important) to 10 (important).

A second step is designed to evaluate already identified factors as they were presented in previous chapters. Here, experts have to allocate points (same scale) to items as well and if so, supplement items of the first step.

A third step is designed to identify the expert's support for the current measurement approach. Here, the experts are faced with the current objective function so that they can decide to support or not support it. Further, they were asked to add or cross out parts within the equation.

D. Demonstration

The previously designed objective function was demonstrated with help of the previously designed expert panel in three workshop dates and the experience of three experts of analytical infrastructures could have been considered.

The authors of the paper moderated the sessions and results are presented based on the following steps:

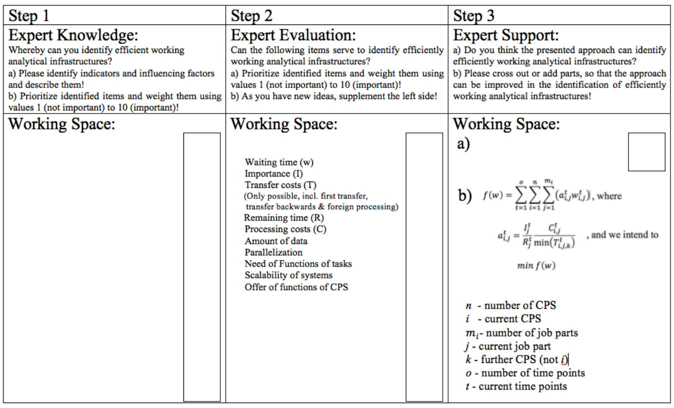

Fig. 2. Layout of an expert panel to validate ojective functions.

During the *Expert Knowledge* identification, indicators and influencing factors for an efficient task processing within analytic infrastructures have been identified and prioritized as group. Those can be seen in Table 1.

During the *Expert Evaluation* identification, objectives were presented, as they were explained in previous sections. As items of the first step have been identified by the experts during the first step (see Table 1), some of them can be connected to items shown in the second step (see Table 2). This connection was discussed and is presented in the following.

Items that could be connected easily were the following: The *capacities* of the entire system and the intention to relate the current system load with a *normed system load* (both in Table 1) underline the identification of bottlenecks. Alternatively, the *waiting time* can be used (Table 2). Further the *processing* and *reaction time* (Table 1) can be expressed as monetary value called *process costs* (Table 2). The *flexibility* in scale, *parallelization* and consideration of *transfer costs* can be identified in both tables.

The following items are considered implicitly: The idea of distributed task processing in the sense of Big Data approaches has been considered with the *role fulfillment* (Table 1). Here, best systems have the lowest *processing costs* (Table 2) as those are used for processing. For this, a *data availability* (Table 1) has to be guaranteed through the distribution through network infrastructures, which is reflected in the use of *transfer costs* (Table 2). Further, the hardware *capabilities* of systems and software *function availability* (Table 1) can be connected to the *offer of functions* (Table 2). Only when a CPS is able to process a certain task, since it knows how because of the software and is able to because of the hardware, processing costs do not become infinite. Then, the *use of functions* because of the realization of a task (Table 1) can be connected to the *need of functions* (Table 2).

The following items were highlighted for a future integration in the current objective function: The consideration of *quality*, *failure safety*, *learning of previous* and *predicting* based on knowledge.

After the discussion, the experts were asked to realize the tasks of the second step. This lead to individual evaluations as they can be found in Table 2.

TABLE I. EXPERT KNOWLEDGE IDENTIFICATION

Item	*Priority*	Item	*Priority*
Transfer costs and transfer inferences	8	Data availability and role fulfillment	6
Processing time, reaction Time	9	Capacities and normed system load	9
Parallelization	6	Quality	9
Function use based on a task evaluation	8	Failure safety	8
Function availability, capabilities	6	Learning of previous, knowledge	6
Flexibility	7	Prediction of system load	10
Data availability and role fulfillment	6		

TABLE II. EXPERT EVALUATION IDENTIFICATION

Item	Expert Evaluation				
	Individual Eval.			*Average*	*Range*
Waiting time	3	5	9	5.67	6
Importance	8	8	6	7.33	2
Transfer costs	7	10	6	7.67	4
Remaining time	10	8	7	8.33	3
Processing costs	10	9	5	8.00	5
Amount of data	5	6	6	5.67	1
Parallelization	7	8	8	7.67	1
Need of functions	7	8	8	7.67	1
Scalability	7	7	7	7.00	0
Offer of functions	5	6	7	6.00	2

Faced with the average and range of the expert evaluation, one can identify the following three variables as most important: *remaining time* (8.33 points) and *processing costs* (8.00 points). The variables *transfer costs*, *parallelization* and *need of functions* share the third place with 7.67 points. Since all variables show greater values than 5.0 points, experts agree in considering them for efficient task processing. None has to be crossed out. The greatest range show the variables *waiting time* (6 points), *processing costs* (5 points) and *transfer costs* (4 points), which can be an indicator for the scenario specific meaning of those KPI. The smallest range show the indicators *scalability* (0 points), task need of function (1 points), *amount of data* (1 points) and *parallelization* (1 points), which can be an indicator for the expert consensus.

During the *Expert Support* identification, a support of all participants could have been identified. Further, the following points were meant to be a sense full supplementation:

- The quality of desired results should be considered as for example less powerful systems can approximate results and return results in time,
- The failure safety, as for example results can get lost or be manipulated violently within Internet similar structures,
- The collection and use of former experiences and based on that, the generation of future predictions can improve the system's task processing,
- The realization of scale effects, so that job parts realize similar operations and can be combined.

E. Evaluation

Objectives have been used in the design of the objective function and expert panel. A demonstration could be realized successfully and gave evidence for a validation.

Faced with the results of the expert knowledge identification of step 1, numerous indicators and influencing factors have been identified by experts that can be mapped to the base of the current objective function. This validates the objective function base. Further ideas have been identified as well.

Faced with the results of the expert evaluation identification of step 2, no item was below a value of 6.0, which supports the current base as well.

As one interprets the expert evaluation based on the average, efficient task processing focus on the *remaining time* of a certain task and as this was possible, allocation algorithms select the CPS with the cheapest *processing costs* and consider task *parallelization* as it was possible because of *transfer costs* and *available functions*.

As one interprets the range of the expert evaluation as agreement of the meaning of indicators, here becomes visible that especially the variables *waiting time*, *processing costs* and *transfer costs* have a different importance, which might because of the selected scenario. Of course, there might be systems that do not care about great waiting times, because their processes are not time dependent. An example could be the communication of a fridge and a microwave that process analytical tasks.

Faced with the results of the expert support of step 3 for the given objective function, none of the experts refused the presented approach. This can be seen as a first validation step. On top of that, further ideas were generated.

IV. CONCLUSION

In this paper, a first approach for a common objective function could have been realized. With this, the research question "*Which common objective function shall be used to optimize task processing within analytical infrastructures?*" could be answered with the objective function of (1) such that an implementation was attractive. For this, the real-world construction of a dynamic, analytical infrastructure processing real-world analytical tasks was needed. Here, as a first step, the measurement should be conducted. Further, the equation should be used for optimization and controlling purposes of analytical infrastructures. After a practical validation, an attempt to generalize the equation for other, non-analytical, systems could be conducted.

On building, the following was attractive: The working of existing Big Data approaches can be compared in using the current objective function as measurement. Here, further allocation strategies can be integrated, such that a benchmarking can be realized concerning actual practices and status of analytical infrastructures in the Internet of Things domain. Hence, attractive strategies can be identified. The objective function can be used as optimization strategy. Further, the integration of ideas identified in the expert panel, was promising. Here, a comparison of different versions of objective functions based on given Big Data approaches was interesting. Of course, a validation on a greater base was attractive as well.

ACKNOWLEDGMENT

The research of A. S. Alfa was mainly supported by the grant from the NRF (South Africa) SARChI Chair in ASN program co-hosted by UP and CSIR.

REFERENCES

[1] B. Bender and M. Grum, "Entwicklung eines Architekturkonzepts zum flexiblen Einsatz von Analytics," INFORMATIK, Lecture Notes in Informatics (LNI), 2016.

[2] K. Peffers, T. Tuunanen, C.E. Gengler, M. Rossi, W. Hui, V. Virtanen, J. Bragge, "The Design Science Research Process: A Model for

Producing and Presenting Information Systems Research," DESRIST, Claremont, CA, 2006.

[3] K. Peffers, T. Tuunanen, M.A. Rothenberger and S. Chatterjee, "A Design Science Research Methodology for Information System Research," Journal of Management Information Systems, Vol. 24, Issue 3, 2007, pp. 45-78.

[4] N. Gronau, M. Grum, B. Bender, "Determining the optimal level of autonomy in cyber-physical production systems," Proceedings of the 14th International Conference on Industrial Informatics Indin, 2016.

[5] B. Vogel-Heuser, G. Kegel, K. Bender, and K. Wucherer, "Global Information Architecture for Industrial Automation," *Automatisierung-stechnische Praxis (atp)*, vol. 51, no. 1, pp. 108–115, 2009.

[6] EMC Education Services: Data Science & Big Data Analytics: Discovering, Analyzing, Visualizing and Presenting Data, John Wiley & Sons Inc. 2015.

[7] Hennessy, J.L.; Patterson D.A.: Computer Architecture - A Quantitative Approach. 4th Edition, 2007.

[8] Hurwitz, J.S.; Kaufman, M.; Bowles, A.: Cognitive Computing and Big Data Analytics, John Wiley & Sons Inc., 2015.

[9] S.R. Lawrence and A.H. Buss, "Shifting production bottlenecks: Causes, cures and conundrums," Journal of Production and Operations Management, Vol. 3, No. 1, 1994, pp. 21-37.

[10] C.T. Kuo, J.T. Lim and S.M. Meerkov, "Bottlenecks in serial production lines: A system-theoretic approach," Mathematical Problems in Engineering, Vol. 2, 1996, pp. 233-276.

[11] S.Y. Chiang, C.T. Kuo and S.M. Meerkov, "Bottlenecks in markovian production lines: A systems aporoach," IEEE Transactions on Robotics and Automation, Vol. 14, No. 2, 1998, pp. 352-359.

[12] A.M. Law and D.W. Kelton, "Simulating modeling & analysis," Mcgraw Hill, 2000.

[13] C. Roser, M. Nakano and M. Tanaka, "Shifting bottleneck detection," Proceedings of the Winter Simulation Conference, 2002, pp. 1079-1086.

[14] P. Taminselvan, K. Krishnan and H. Cheraghi, "Measurement of shifting bottlenecks in a non-buffer production system," Proceedings of the Industrial Engineering Research Conference, Cancun Mexico, 2010.

[15] N. Chiadamrong and P. Limpasontipong, „Using Storage Buffer to Improve Unbalances Asynchronous Production Flow Line's Performance," International Journal for Manufacturing Technology and Management, Vol. 5, 2003, pp. 149-161.

[16] P. Tamilselvan, K. Krishnan and H. Cheraghi, „Measurement of shifting bottlenecks in a non-buffer production system," Proceedings of the Industrial Engineering Research Conference, Cancun Mexico, 2010.