SPS2020
K. Säfsten and F. Elgh (Eds.)

doi:10.3233/ATDE200157

Optimizing Ergonomics and Productivity by Connecting Digital Human Modeling and Production Flow Simulation Software

Aitor IRIONDO PASCUAL[a,1], Dan HÖGBERG[a], Anna SYBERFELDT[a], Erik BROLIN[a] and Lars HANSON[a,b]
[a] *School of Engineering Science, University of Skövde, Skövde, Sweden*
[b] *Scania CV, Södertälje, Sweden*

Abstract. Simulation software is used in the production development process to simulate production and predict behaviors, calculate times, and plan production in advance. Digital human modeling (DHM) software is used to simulate humans working in production and assess whether workstation designs offer appropriate ergonomic conditions for the workers. However, these human simulations are usually carried out by human factors engineers or ergonomists, whereas the production simulations are carried out by production engineers. Lack of integration of these two simulations can lead to suboptimal solutions when the factory is not optimized to improve both productivity and ergonomics. To tackle this problem, a platform has been developed that connects production flow simulation software data and DHM software data and integrates them in a generic software for data treatment and visualization. Production flow simulation software data and DHM software data are organized in a hierarchical structure that allows synchronization between the production data and the ergonomic data on the target simulation software. The platform is generic and can be connected to any production flow simulation software and any DHM software by creating specific links for each software. The platform requires only the models of the production line, workstations, and tasks in order to perform the simulations in the target simulation software and collect the simulation results to present the results to the user of the platform.

Keywords. Production, Ergonomics, Simulation, DHM, Optimization

Introduction

There are many ways to optimize a factory, including creating new workstations, changing the order of assembly operations, balancing the tasks of each workstation, and increasing or decreasing the buffers [1]. The full range of possible configurations cannot be tested in a real factory given the time it would take to make these changes and the associated costs. Consequently, software has been developed to simulate production in a factory, allowing different configurations to be tested and analyzed without changing the actual factory. Such software is very useful when redesigning a factory. It can simulate the behavior of a new configuration, allowing production engineers to optimize the configuration through several iterations, testing and observing the main production factors such as throughput, work in process, and lead time [2]. Examples of production

[1] Corresponding Author. aitor.iriondo.pascual@his.se

simulation software include FlexSim [3], Siemens Tecnomatix Plant Simulation [4], and FACTS Analyzer [5].

However, in order to simulate a factory that requires workers, it is also necessary to simulate human factory workers. Digital human modeling (DHM) software can be used for this purpose [6]. The simulated humans, commonly called digital manikins or just manikins, can be analyzed in terms of ergonomics. DHM software thus enables products and systems, such as workstations, to be analyzed from a human-centered perspective in a virtual world. It enables an efficient iterative design process to find suitable solutions. Hence the use of DHM software matches the purpose of production simulation software, but DHM software focuses on simulating human work. There are numerous DHM programs to perform human simulations and evaluate ergonomics, including Siemens Jack [7], Santos [8], and IPS IMMA [9]. However, these human simulations are usually carried out by human factors engineers or ergonomists to validate workstations, without integrating these simulations with those performed by production engineers. As a result, simulations performed to predict production are usually done separately from human simulations performed to evaluate ergonomics. This can lead to suboptimal design solutions from a combined productivity and ergonomics perspective.

The main focus in this paper is on developing a method to connect production flow simulation software and DHM software in a common platform. The purpose is to enable concurrent consideration of production and ergonomics, and to establish a foundation for performing simulation-based multi-objective optimization of productivity and ergonomics. In this way, the process of performing multi-objective optimization of ergonomics and productivity can be automated, so reducing redesign times and enabling more optimized design solutions to be found.

1. Method

In information systems (IS) research, the design and creation methodology defines the steps involved in developing and evaluating an artifact, which may be a construct, model, method, instantiation, or framework [10] (Figure 1). The validation of the artifact corresponds to the criteria evaluated in each iteration of the artifact design. In this paper, design and creation methodology is applied to the development and evaluation of a method and a platform (i.e., an instantiation) to enable connection of simulations from DHM software and production flow simulation software. This platform will allow simulation-based multi-objective optimizations of ergonomics and productivity simultaneously. The evaluation criteria for the method and the platform are ensuring that both ergonomics and productivity data are available in the same data structure and that the data is suitable for optimization.

Figure 2 depicts how the design and creation methodology was adapted and used to develop and evaluate the platform.

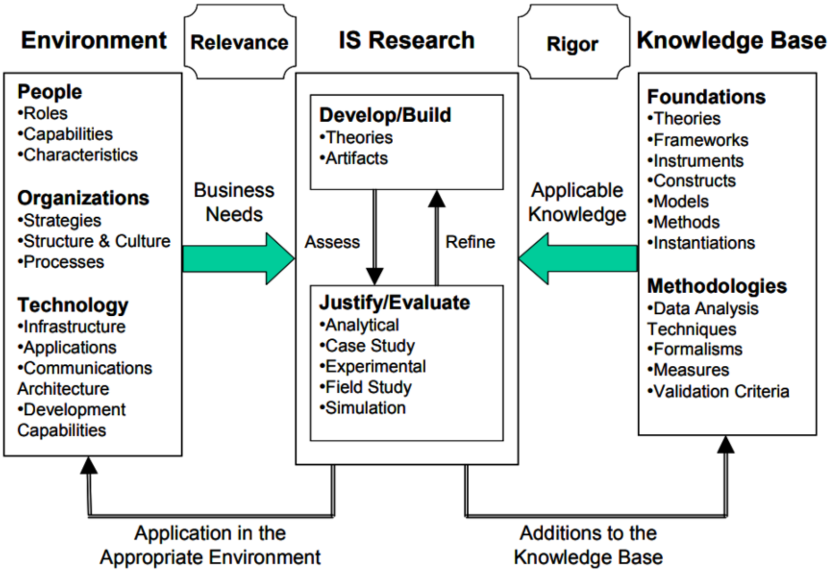

Figure 1. Design and creation methodology framework [11]

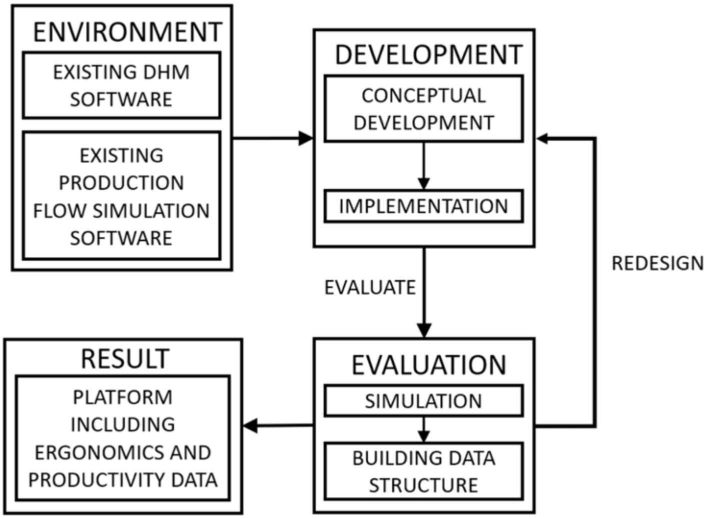

Figure 2. Methodology used to develop and evaluate the platform

This research was carried out in a technological environment where the platform being developed is to function as a link between DHM simulation software and the production flow simulation software currently used in industry (Figure 2). Therefore, the first step was to study both these types of software from a generic viewpoint in order to understand the information that the platform must be able to process and contain. After this, the research process for the development of the platform was initiated.

The development of the platform was an iterative process of development and evaluation through simulation (Figure 2). The development was based on a two-step process, where the connections necessary to communicate with DHM simulation software and production flow software were conceptually defined, and then implemented in code. This platform implementation was used as a link between DHM and production flow simulations and was evaluated by simulating the process of integrating the information of the DHM and the production flow simulation software. The criterion for evaluating the process was whether the ergonomic information of the DHM software and the productivity information of the production flow simulation software could be included in a single information structure that combines both ergonomics and productivity.

Following a two-step process, the research was divided into two parts: the conceptual development of the platform through the study of DHM simulation and production flow software, and the practical implementation of the platform. This in accordance with the design and creation process of developing an artifact and evaluating it through simulations (Figure 2). This iterative process of conceptual development, implementation, and evaluation through simulations was continued until an information structure that included both ergonomics and productivity information was obtained.

2. Conceptual model of the platform

In order to connect DHM and production flow simulation software, it is necessary to know what inputs DHM and production flow simulation software typically require and what output they typically provide.

2.1. Production flow simulation software

The typical input for production flow simulation software consists of a model of the production line created by the user within the software. This model includes the elements that make up the production line, for which the production flow simulation software offers elements such as workstations, buffers, warehouses, conveyor belts, material entries, and product demand. These elements are modifiable; for example, workstation times, standard deviations of production, repair times, and other variables can be defined. Once all the elements have been defined, the production line can be simulated. The result of the simulation provides output data on the complete production line and each element in it, enabling assessment of the throughput of the line, the percentage of occupation of the buffers, and other production metrics.

2.2. DHM software

The typical input in DHM software is a 3D model of the environment. The development of the platform is based on the assumption that the DHM software can provide manikin motions using an algebraic or data-driven approach. In the 3D environment, the manikins are defined and the tasks to be performed are specified. A simulation then generates the motions of the manikins to perform the tasks as output. These motions are represented on a timeline, so the time the manikins need to complete the tasks can be obtained. In addition, ergonomic evaluations can automatically be performed, assessing the risks for work-related musculoskeletal disorders (WMSDs).

2.3. Data hierarchy

In order to be able to connect the data from the production flow simulation software with data from the DHM software, a data structure was defined in which the data is organized by levels, establishing a hierarchy between these levels. This hierarchy allows the establishment of a data flow so that the upper levels depend on the data of the lower levels (Figure 3). In this case, (1) the production line is at the highest level, since it is composed of the rest of the elements. The production elements are at a lower level; in this case these are (2) the workstations. Under the workstations are (3) the tasks, and (4) the manikins that perform the tasks.

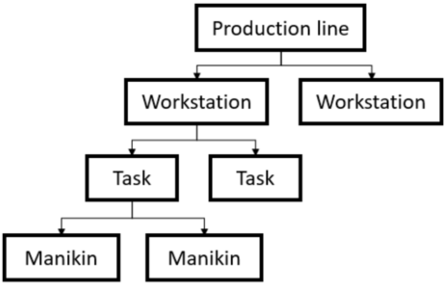

Figure 3. Data hierarchy

These levels were defined by the level of detail and by the dependence of the data in the simulation software. In the real world, humans perform various tasks and therefore humans would be at a higher level than tasks at the organizational level. However, in simulation software manikins are typically assigned to tasks. In some cases several manikins are assigned to a single task. This data dependency is why the manikins have been placed below the tasks in the hierarchy of data.

2.4. Communication with simulation software

A platform was created (Figure 4) in order to establish this hierarchy and to be able to connect the two simulation software programs. This platform includes data from both types of software and is able to communicate with both programs. The platform is also responsible for creating hierarchical data levels when communicating with software, so that the data generated by the simulation software is matched with the predefined hierarchical data levels.

2.5. Communication links

The objective was to make the platform as generic as possible so that it can communicate with any production flow simulation software and with any DHM software. To enable this, links will be created for each software connected to the platform (Figure 4). The links will take the data from the target simulation software and enter it into the platform, organizing the data in a hierarchical structure. Each link will be unique to a particular software program, since access to the data of each software is performed differently, either by communication protocols or involving different types of files, in addition to having a unique data structure.

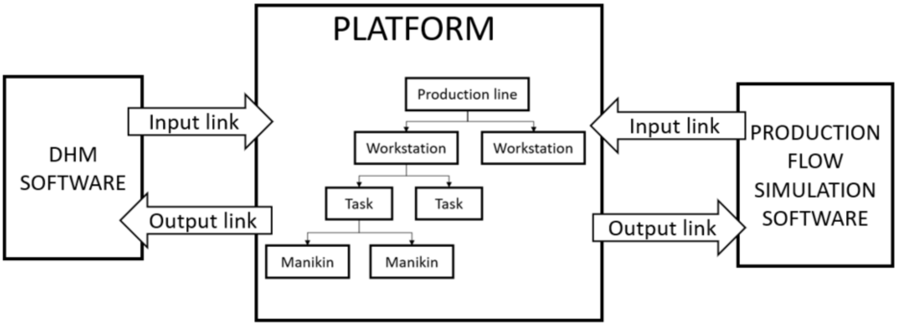

Figure 4. Platform communication with DHM and production flow simulation software

2.6. Workflow of the platform

The platform requires a specific workflow to input the data in order to be able to keep the data synchronized in a hierarchical manner. As this hierarchical structure has the production line at the highest level, it is necessary to define the production line first to be able to locate the rest of the elements below it, so that the production line is the root and branches into the other elements. Therefore, the input of the data should be from the highest level to the lowest level, with a top-down approach so that the data structure is built by defining the production line, then the workstations, and then the tasks to be performed by the manikins. The platform performs the simulations in the opposite direction, from the lowest levels to the highest levels in a bottom-up approach, since each level is dependent on the data of the lower level (Figure 5).

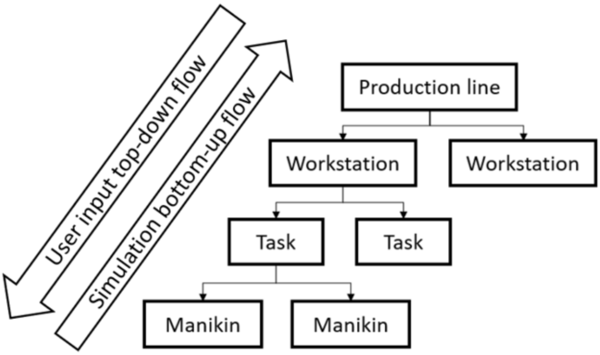

Figure 5. User input and simulation workflow

3. Platform implementation

The result is a platform that allows DHM simulation software to be connected with production flow simulation software. The steps taken to complete this platform are described below.

3.1. The platform

A platform was created using C++ as the main programming language. This language was chosen because of its object orientation, standardization, strong typing, and processing efficiency [12]. In addition, the platform was developed within the QT Creator environment, so that the platform interface is based on the graphic elements that QT Creator offers. These graphic elements are also considered C++ objects [13].

The use of object-oriented programming allows the data entered in the platform to be organized as objects. In this case the main objects are (1) the production line, (2) workstations, (3) tasks, and (4) manikins. Each of these objects can contain several objects of the underlying types. In this way a hierarchy is generated between these objects, so that a (1) production line can contain several (2) workstations, (2) workstations can have several (3) tasks, and (3) tasks can be performed by several (4) manikins.

In addition, each main object contains secondary objects that contain the data of the main object. For example, in the case of a manikin, the secondary objects are anthropometrics, joint positions, ergonomic evaluations, and settings (Figure 6). These secondary objects make it possible to organize the data of the main objects so that it is easier to locate and use, while allowing the data to be entered in different ways for each secondary object. For example, while repair times of a machine may be given by the production simulation, task times can be defined by the time it takes for a manikin to complete a task.

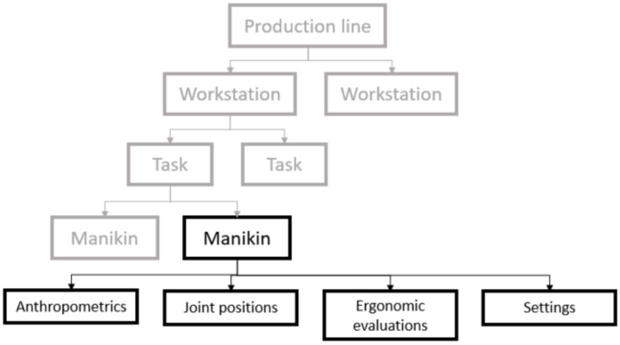

Figure 6. C++ object hierarchy with objects of a manikin

3.2. Connecting the platform to production and ergonomics simulation software

To demonstrate and evaluate the created method and platform, the production flow simulation software FACTS Analyzer [5] and the DHM software IPS IMMA [9] were connected to the platform. The FACTS Analyzer models are defined in XML files. The simulation output, indicating the production factors of the line and each station, can be obtained in TXT format or in XML format. The XML format was chosen as the data transfer format to be able to organize the data in a more structured way than the TXT format offers. The platform reads the FACTS Analyzer model XML, and automatically runs the line simulation and reads the line XML output to obtain the productivity results (Figure 7).

IPS IMMA was connected to the platform through communication via JSON files (Figure 7). In this case, the IMMA IPS model files are more complicated since they contain 3D model data. The data of the models and the simulation results are not directly accessible in the files, as in the case of the FACTS Analyzer XML files. IPS IMMA does not natively offer an export or import of data in a standard file type. In order to perform this export, IMMA offers a scripting language called Lua Scripting, through which the data of the models can be accessed through code [14]. Therefore, two codes in Lua Scripting were developed. The first code exports data from IPS IMMA to the platform, and the second code imports data to IPS IMMA. The code to export data creates JSON files with all the necessary data for the platform, and the import code reads the JSON files in order to make changes to the models. JSON format was chosen for the file type due to the speed of reading and writing, as well as the ease of writing long numerical arrays. This was not necessary in the communication with FACTS Analyzer since the amount of numerical data was small enough. However, IPS IMMA provides the vectors of each joint position of the simulated manikin for each simulated frame, which can result in large amounts of numerical data.

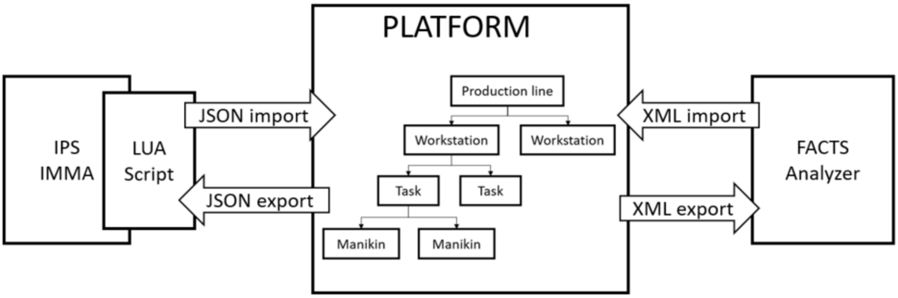

Figure 7. Platform communication with IPS IMMA and FACTS Analyzer

3.3. Simulation and data transferring workflows

The first test of the platform used a top-down workflow for the user input and a bottom-up workflow for the simulations (Figure 5). At the outset, a model of the production line was created in FACTS Analyzer, so that the material input, work stations, and buffers prior to each workstation were defined (Figure 8). This model was saved in XML format.

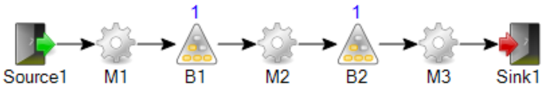

Figure 8. Example of a model in FACTS Analyzer

Subsequently, the created platform was executed and the FACTS Analyzer model was loaded into it. The user input required was the IPS IMMA export files corresponding to each workstation of the FACTS Analyzer model. Therefore, the next step was to open IPS IMMA and create 3D models of each workstation (Figure 9) and the tasks of the manikins, and then export these models to the platform. When exporting the models of the workstations, the tasks were executed automatically and the IPS IMMA JSON export files generated by Lua scripting were imported into the platform. In this case, these JSON export files provided the platform with data about the ergonomics of the manikins and the predicted times to perform each task.

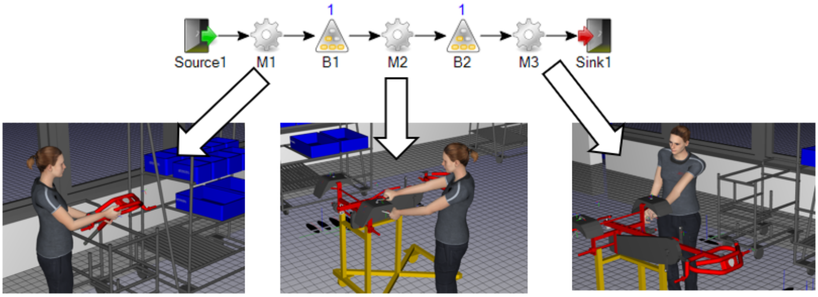

Figure 9. IPS IMMA models representing FACTS Analyzer workstations

Once the platform contains the models from the FACTS Analyzer and IPS IMMA, it generates the corresponding objects from the production line, workstations, tasks, and manikins. The platform then simulates the tasks in IPS IMMA. If desired, the times of the human simulations can be used to adjust the times of the workstations of the production line, to finally automatically execute the simulation in FACTS Analyzer. This provides the platform with the results of the production simulation with adjusted production times from the human simulations.

3.4. Resulting platform and data files

After having generated all the data within the platform, the platform continually synchronizes the production line model created in FACTS Analyzer together with the IPS IMMA models. Therefore, all the data can be accessed from the platform. This means that both the production line with its production metrics such as throughput and the ergonomic evaluations of the workers who perform the tasks at each workstation are available. In addition, the data is synchronized so that human simulations define production times, affecting different production metrics. If the production line model is modified, the platform asks the user for the necessary changes to ensure synchronization. For example, if a new station is added to FACTS Analyzer, the platform will request the IPS IMMA export files to be able to define the workstation processing times. In turn, every time a change is made to an IPS IMMA model, the platform will automatically run the simulation in FACTS Analyzer to ensure that the simulation of the production line reflects these changes.

The production line model with workstations, tasks and manikins is temporarily stored in the platform's memory, so that it can be used during the session. This data can be saved in files, but instead of using a single file for the entire model, several files are generated so that the files do not become too large. The production line object is saved in a JSON file (Figure 10) that indicates which workstations it includes. A JSON file is also generated for each workstation, indicating what tasks it includes. Finally, a JSON file is generated for each task together with the data of the manikins that perform each task, containing data of manikin motions, anthropometrics, and ergonomic evaluations. Creating a file for each task facilitates reading and modification of each file, so that the processing times for synchronizations decrease. The use of JSON files to save all the results of the platform allows any program to read them, making the platform an open environment that can provide both production and ergonomic data to other software, and can even be read by a user without the need to have access to the platform.

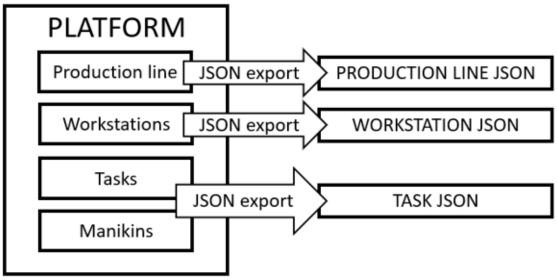

Figure 10. JSON exports divided into hierarchical levels

4. Discussion

The platform described enables a connection between human and production simulations. This allows the implementation of algorithms to change the design of production lines and workstations and to balance production lines. It also opens up the possibility of multi-objective optimizations in order to concurrently improve productivity and ergonomics.

It is anticipated that the platform will also enhance understanding and collaboration between production engineers and ergonomists in the production development process. Thus, for example, the platform can highlight and facilitate the consideration of ergonomics when designing production lines or workstations.

The platform's ability to consider ergonomics depends largely on the functionality provided in the DHM software. Current DHM software typically represents only physical aspects of humans, such as posture, motions, fit, reach, biomechanical loads, and risks for WMSDs and neglects other issues or domains of ergonomics such as temperature, noise, cognitive ergonomics, and organizational ergonomics. If information on these issues is not provided by the DHM software, some of the issues can be implemented in the platform by manual input from the user. There is ongoing research on including aspects of cognitive science and ergonomics in DHM software, for example, in relation to the selection of actions [15]. When such functionality is available in DHM software, the objective is to enable the platform to consider more aspects of ergonomics.

Clearly defining what is meant by improving and optimizing ergonomics is not a straightforward matter. Some studies have tried to solve this problem by analyzing different ergonomic evaluation methods [16]. However, the common approach when optimizing ergonomics is to consider results from different ergonomic evaluation methods, given as scores of the risks for WMSDs, and to optimize for decimals of the score [17]. It is questionable whether such approaches are really helpful for representing the risk of WMSDs, especially when methods developed for the assessment of static instances are used to assess work sequences, for example, in time-based simulations.

Time is an essential consideration in ergonomic assessments [16], [18], [19]. Thus the platform needs to be able to consider outcomes from time-based ergonomic evaluation methods. To elaborate, from an optimization point of view, it might be desirable to discretize ergonomic loads and associate these loads with smaller blocks of tasks. However, this might be problematic from an ergonomic point of view if ergonomic evaluation methods are developed to assess accumulated biomechanical loads, for example, a full workday's accumulated loads. There is a need for the creation or adaption of ergonomic evaluation methods that can be used in optimization and that consider ergonomics in a representative way. These challenges need to be addressed in future research in the development of the method and platform introduced in this paper.

Acknowledgment

This work has been made possible with support from ITEA3 in the project MOSIM, and with the support from the Knowledge Foundation and the associated INFINIT research environment at the University of Skövde, within the Virtual Factories – Knowledge-Driven Optimization (VF-KDO) research profile and the Synergy Virtual Ergonomics (SVE) project, and by the participating organizations. This support is gratefully acknowledged.

References

[1] I. Siregar, J. B. Saedon, M. S. Adenan, S. Shawal, and M. F. Othman, Modelling and Analysis of Manufacturing Process Layout, IOP Conf. Ser. Mater. Sci. Eng., vol. 598, p. 012051, Sep. 2019, doi: 10.1088/1757-899X/598/1/012051.

[2] D. Górnicka, J. Kochańska, and A. Burduk, Production Resources Utilization Improvement with the Use of Simulation Modelling, in Information Systems Architecture and Technology: Proceedings of 40th Anniversary International Conference on Information Systems Architecture and Technology – ISAT 2019, 2020, pp. 41–50, doi: 10.1007/978-3-030-30604-5_4.

[3] Y. Shi and J.-J. Wang, Optimization of Production System by Simulation Based on UML and Flexsim, in 2011 International Conference on Management and Service Science, Aug. 2011, pp. 1–4, doi: 10.1109/ICMSS.2011.5999204.

[4] J. Siderska, Application of Tecnomatix Plant Simulation for Modeling Production and Logistics Processes, Bus. Manag. Educ., vol. 14, no. 1, pp. 64–73, 2016.

[5] A. Ng, M. Urenda Moris, J. Svensson, A. Skoogh, and B. Johansson, FACTS Analyser : An innovative tool for factory conceptual design using simulation, presented at the The Swedish Production Symposium, Gothenburg, August 28-30, 2007, 2007, Accessed: Oct. 16, 2019. [Online]. Available: http://urn.kb.se/resolve?urn=urn:nbn:se:his:diva-7334.

[6] S. Scataglini and G. Paul, DHM and Posturography. Elsevier Academic Press, 2019.

[7] D. P. Boros and K. Hercegfi, Digital Human Modelling in Research and Development – A State of the Art Comparison of Software, in Human Systems Engineering and Design II, 2020, pp. 543–548.

[8] E. P. Pitarch, J. Yang, and K. Abdel-Malek, SANTOS Hand: A 25 Degree-of-Freedom Model, presented at the 2005 Digital Human Modeling for Design and Engineering Symposium, Jun. 2005, pp. 2005-01–2727, doi: 10.4271/2005-01-2727.

[9] D. Högberg, L. Hanson, R. Bohlin, and J. S. Carlson, Creating and shaping the DHM tool IMMA for ergonomic product and production design, Int. J. Digit. Hum., vol. 1, no. 2, pp. 132–152, Jan. 2016, doi: 10.1504/IJDH.2016.077413.

[10] S. T. March and G. F. Smith, Design and natural science research on information technology, Decis. Support Syst., vol. 15, no. 4, pp. 251–266, Dec. 1995, doi: 10.1016/0167-9236(94)00041-2.

[11] A. R. Hevner, S. T. March, J. Park, and S. Ram, Design Science in Information Systems Research, MIS Q, vol. 28, no. 1, pp. 75–105, Mar. 2004.

[12] M. Sakkinen, Inheritance and other main principles of C++ and other object-oriented languages, Jan. 1992.

[13] K. Hartness, Graphics and user interfaces in C++ with Qt, J. Comput. Sci. Coll., vol. 20, no. 4, pp. 198–199, Apr. 2005.

[14] A. M. Maidl, F. Mascarenhas, and R. Ierusalimschy, A formalization of typed lua, in Proceedings of the 11th Symposium on Dynamic Languages, Pittsburgh, PA, USA, Oct. 2015, pp. 13–25, doi: 10.1145/2816707.2816709.

[15] E. Billing, L. Hanson, M. Lamb, and D. Högberg, Digital Human Modelling in Action, presented at the SweCog 2019, the 15th SweCog conference, Umeå, Sweden, November 7-8, 2019, 2019, pp. 25–28, Accessed: Nov. 29, 2019. [Online]. Available: http://urn.kb.se/resolve?urn=urn:nbn:se:his:diva-17882.

[16] C. Berlin and T. Kajaks, Time-related ergonomics evaluation for DHMs: a literature review, Int. J. Hum. Factors Model. Simul., vol. 1, no. 4, pp. 356–379, 2010, doi: 10.1504/IJHFMS.2010.040271.

[17] M. Bortolini, M. Faccio, M. Gamberi, and F. Pilati, Multi-objective assembly line balancing considering component picking and ergonomic risk, Comput. Ind. Eng., vol. 112, pp. 348–367, Oct. 2017, doi: 10.1016/j.cie.2017.08.029.

[18] R. Wells, S. E. Mathiassen, L. Medbo, and J. Winkel, Time—A key issue for musculoskeletal health and manufacturing, Appl. Ergon., vol. 38, no. 6, pp. 733–744, Nov. 2007, doi: 10.1016/j.apergo.2006.12.003.

[19] I. M. Rhén et al., Ergonomic risk assessment in DHM tools employing motion data - exposure calculation and comparison to epidemiological reference data, Int. J. Hum. Factors Model. Simul., vol. 6, no. 1, p. 31, 2018, doi: 10.1504/IJHFMS.2018.091356.