Collaborative Design of Material Handling Systems Using Distributed Virtual Reality Environments

Orthodoxos Kipouridis[1(✉)], Moritz Roidl[2], Marcus Röschinger[1], Michael ten Hompel[2], and Willibald A. Günthner[1]

[1] Institute for Materials Handling, Material Flow, Logistics, Technische Universität München, Munich, Germany
{kipouridis,roeschinger,kontakt}@fml.mw.tum.de

[2] Chair of Materials Handling and Warehousing, Technische Universität Dortmund, Dortmund, Germany
{moritz.roidl,michael.tenHompel}@tu-dortmund.de

Abstract. In order to integrate with an upcoming ecosystem of interconnected products and services, adaptive, intelligent and autonomous material handling systems (MHS) are being deployed in modern logistics facilities, enabling them to cope with requirements for high flexibility, configurability and reusability. Recent advances in virtual engineering and virtual reality (VR) technologies offer new possibilities for remote collaboration between companies regarding the design process of MHS, which involves several stakeholders. In this paper, we present a 3D collaborative virtual environment (3DCVE) as part of distributed multi-platform software for supporting MHS design tasks in logistics facilities. We describe how an in-memory data grid technology can be used for the distribution of data and the enabling of consistent user access to the virtual environment (VE). The main focus is put on the use-case of facility layout planning presenting the implementation of approaches regarding user interaction and collaboration.

Keywords: 3D collaborative virtual environments · Material handling systems design · Virtual reality

1 Introduction

Increasingly shorter product life cycles, growing product version ranges as well as more complex and global value chains pose new challenges to manufacturing and logistics. Against this background, industry and research have identified several courses of action which are often included in the frameworks of Industry 4.0 and the Industrial Internet. These aim for higher flexibility, changeability and efficiency within industrial processes. A promising approach is the introduction of intelligent, interconnected manufacturing and material handling systems (MHS) on the shop floor. Such systems -also called cyber physical systems (CPS), usually refer to embedded systems that can communicate with each other, targeting seamless integration towards an internet of things (IoT) and services. Because of their intelligent and decentralized control, CPS's are able to make

S. Lackey and R. Shumaker (Eds.): VAMR 2016, LNCS 9740, pp. 609–620, 2016.
DOI: 10.1007/978-3-319-39907-2_58

decisions autonomously and locally within a complex system and therefore increase flexibility and reactivity [1].

Enabled by advances in the fields of automation and communication, CPS's have also been introduced in materials handling. Initial industrial deployments have been brought to market as flexible conveyor belts, self-guided transport systems and autonomous, human-interacting robots [2]. Despite the advantages like high flexibility and reusability offered by these systems, their complexity, varied communication interfaces and lack of standardization have posed challenges to planning logistics facilities, particularly in the phases of system design and integration. Furthermore, the involvement of several stakeholders, such as facility and logistics planners, manufacturers of MHS's, system integrators and facility owners, complicates the planning and integration process.

We present a virtual engineering concept utilizing distributed 3DCVEs, which extends the work presented in [3] regarding a cloud-based software platform, with a view to support an integrated, effective and remote collaboration between the parties involved in the planning phase of heterogeneous MHS's. Furthermore, it is shown how planning activities, such as the plant layout design, visualization and testing of emulated and real MHS can be synchronized and collaboratively edited by different users.

This paper is structured as follows: it initially introduces virtual engineering and an overview of current approaches regarding collaborative planning of technical systems, investigating the potential of using VE's for collaboration in logistics facility planning. It then discusses the concept and its implementation details, demonstrating how an in-memory cloud-technology, primarily used for caching, can be used to enable distributed 3DCVEs. Following, we present our approach to topics like handling user interactions, transmission of data and virtual presence. Finally, it discusses the challenges and potential of the proposed concept, considering its generic aspects and portability with respect to other use cases outside the area of MHS.

2 Related Work

This chapter discusses the concepts of virtual engineering, its main methods and applications. In the second part, it provides an overview of the planning process for logistics facilities. Finally, current approaches regarding integration of virtual engineering in the planning process are presented.

2.1 Virtual and Collaborative Engineering

The term *virtual engineering* covers all information, communication, and visualization technologies that are used in a combined effort to optimize and accelerate the product development and production planning process [4, 5]. It covers three design elements, namely *concurrent*, *virtual* and *collaborative engineering*. Whereas concurrent engineering refers more to organizational aspects, enabling the parallel processing of traditionally successive design steps, in order to detect design or production problems earlier. Virtual and collaborative engineering focuses more technical concepts and are therefore subsequently described more in detail.

Virtual engineering applications incorporate computer graphics, sound, and networking to simulate the experience of real-time interaction in a shared, 3D virtual world. Virtual prototypes can be produced, reviewed and optimized in the early stages of the development process. In a multi-user environment, each user runs an interactive interface program on a client computer connected to a wide-area network (WAN). The interface program simulates the experience of immersion in a VE by rendering images and audio of the environment as perceived from the user's simulated viewpoint. Each user is represented in the shared VE by an entity rendered on every other user's computer, called an *avatar*. Multi-user interaction is supported by matching user actions to entity updates (e.g. motion and sound generation) in the shared VE.

Collaborative engineering denotes the cooperation of various actors, e.g. manufacturers, suppliers, customers and service providers in distributed networks with different IT infrastructures for development, planning and implementation projects. Most of the processes are supported by internet technologies and actors collectively work on tasks within a distributed environment. Collaborative engineering aims at the joint development of technological and competitive advantages which would not be achieved by single individuals. In order to achieve its goals, collaborative engineering requires intensive communication between partners and appropriate forms of organization to control inter-company workgroups. ICT products, and associated human resource activities, such as coaching, can be accommodated to increase the effectiveness of collaborative engineering process [6].

2.2 Collaborative Design of Material Handling Systems

There have been various methodologies proposed for planning warehouses and factories that include intelligent systems in the area of manufacturing and logistics [7, 8]. The following focuses on planning the deployment of MHS for logistics facilities.

In a typical logistics facility, a large number of software and hardware systems have to interoperate in order to accomplish tasks, such as storing, picking and transporting different goods. MHS, such as conveyors, picking systems, automated guided vehicles (AGV) and automated storage and retrieval system (AS/RS), include a large number of sensors, actuators and mechanical components as well as interfaces to software systems, like enterprise resource planning (ERP), and warehouse management systems (WMS) as shown in Fig. 1.

During the MHS design process various partners are required to work closely together in order to ensure the functionality and interoperability of the diverse system elements. In doing so, a large amount of data needs to be exchanged and reconciled, for which a variety of mediums and systems is used. Thus, the consistency and transparency of data, which are necessary for efficient and reliable decisions, can hardly be guaranteed. This inconsistency leads not only to further communication efforts, but also to additional time and monetary cost. Virtual engineering can address this need for collaboration by providing all stakeholders with suitable tools and techniques.

The design process of a heterogeneous MHS for a logistics facility is an iterative process, in which a simplified version of it is described in Fig. 2. In the preparation phase, the project stakeholders (e.g. systems manufacturers, clients and planners) are

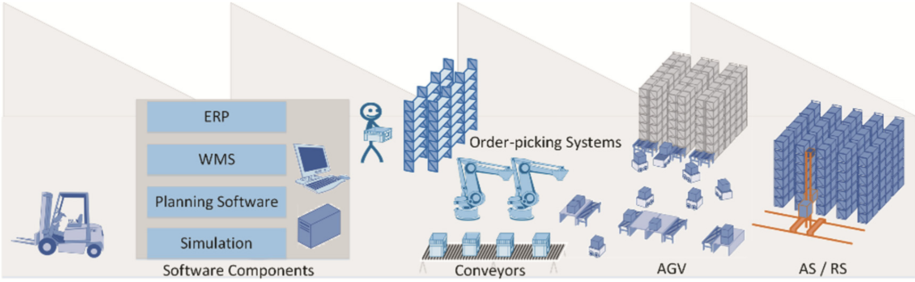

Fig. 1. Components of a modern logistics facility

responsible for generating the relevant project data, such as 3D models, interface specifications and building plans, for their respective systems, In the second step, the partners work together to create a rough general plan of the complete system under deployment. In the detailed planning phase, the communication interfaces between the systems have to be configured so that they can exchange information and interoperate. Finally, in the realization planning phase, the real systems have to be tested by executing test scenarios in order to prove their functionality, reliability and safety.

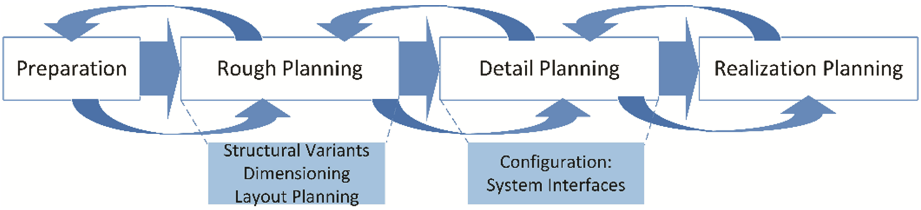

Fig. 2. Design Process of MHS (based on [9])

The concept proposed in this paper focuses on the rough planning phase, which also deals with the layout planning. With a view to create structural variants, the positions of systems are to be determined taking into consideration facility layout plans and previous installments. Current layout planning practices involve using a variety of CAD software as well as paper-printed layout plans. All these digital and non-digital plans are accessed and edited in a serial fashion by respective users, creating multiple versions of these plans. The documents are subsequently shared using conventional (e-mails) and modern methods (cloud shared folders).

2.3 Related Approaches for Collaborative Virtual Environments

In the past, various approaches and frameworks have been proposed to creating multi-user collaborative VE (such as those presented in [10, 11]), while recently, a number of single user software offerings, mostly involving CAD and product lifecycle management software (PLM), incorporate multi-user collaboration features. VR-Systems can in general be categorized as immersive (e.g. Cave Automatic Virtual Environment - CAVE), semi-immersive (Head Mounted Display - HMD) and non-immersive (displays). An extended review of the available desktop VR systems technologies is given in [12].

Following the advances in the VR, several tools have been proposed that target the integration of VR technology into the factory planning process. Examples of such efforts can be found in [13–15]. However, these approaches aim at factory planning processes, which, despite similarities with logistics facilities, do not specifically target heterogeneous MHS planning.

There is a number of challenges that are common to collaborative software engineering. Namely, following issues have been identified as most commonly occurring in collaborative software engineering tasks and should be avoided by newly developed engineering processes [16]:

- Storing information in different systems and at different locations
- Using different or incompatible formats of planning related files
- Lack of knowledge sharing mechanisms
- Parallel processing of different document versions.

With a view to address these challenges for the use case of supporting an enterprise-wide planning and realization of MHS, a multi-user distributed 3DCVE is presented in the following section.

3 Approach for a Multi-user, Distributed 3DCVE

As highlighted in the previous chapter, the design of modern MHS is a rather complex task that requires a high level of cooperation and awareness from every partner involved in the planning project.

With a goal to address the need for an effective, remote collaboration between stakeholders, a distributed software platform, named KoDeMat [17], has been developed. It supports designing and testing of decentralized controlled, heterogenous MHS in logistics facilities. The platform aims not only to support traditional planning processes, but also to enrich them with collaborative ones. It is based on a 3-tier, service-oriented architecture, as shown in Fig. 3, and is scalable to offer the provision of software-as-a-service according to project requirements.

In order to enable the collaboration between partners, a networking layer is required to distribute data between users. Based on the data distribution layer, the middle service-layer provides services that can support tasks in the MFS design process (e.g. interface definition, plant layout design and visualization of emulated and real systems). One of the main collaboration services of the platform is the 3DCVE. The services are made available to users through the presentation layer that includes topics such as interaction of users with other users as well as with the VE. The following paragraphs present the implementation, functionality and challenges of 3DCVE.

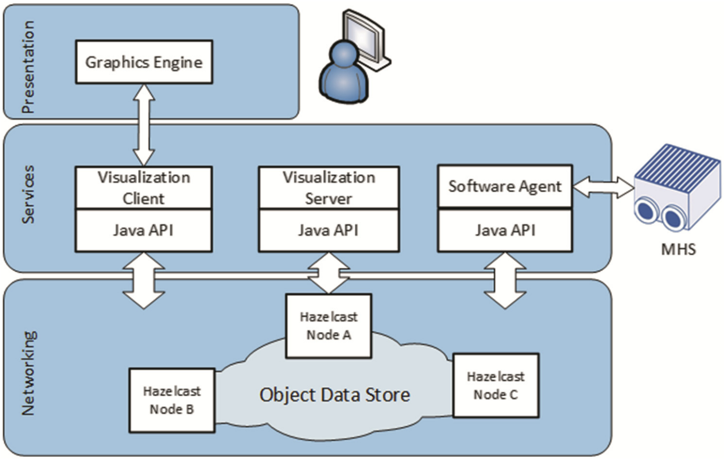

Fig. 3. Platform architecture for enabling distributed VE

3.1 Using an In-Memory Data Grid for Enabling Distributed VEs

A prerequisite to enable the synchronous collaboration of multiple remote users is the distribution of data. The proposed networking scheme is based on a concept of a data-centric, grid-based object distribution. The main idea of this concept is to deploy a NoSQL data store as a partitioned storage system that distributes data among nodes in a cluster. A similar approach of using a data-centric approach to coordinate distributed access to VE is presented in [18], though the networking layer differs from our concept as it uses a hierarchical peer-to-peer networking for the data store.

The KoDeMat platform introduces a networking scheme based on a tiered, object-sharing architecture that employs a framework for distributed computing, *Hazelcast*, which is primarily used in high-performance cloud computing. Hazelcast [19] addresses the transactional consistency as well as object serialization and duplication. Although in most cases transactional approaches are not favored for highly dynamic applications like visualization due to the increased message overhead that they introduce, this is not the case with the selected distribution technology, as it can support thousands of connected nodes with low latency. The drafted networking architecture is shown at the bottom layer of Fig. 3. A client plus member topology was selected as described in [20]. Every user in the network runs a node agent in a java virtual machine (JVM), which ensures the consistency and redundancy of the data.

In this design, users can join an ongoing session in the VE at any time. As soon as a new user joins, his client has access to the current state of objects that determine the state of the scene. The data distribution architecture ensures that all objects in the network have the same state. Similarly, if users disconnect (leave the session or crash), their data are not lost.

The process which communicates object changes to the graphics engine is executed in a separate thread (visualization server) that observes changes and updates the scene accordingly. Consequently, the proposed concept allows for cross-platform user interaction, independently of the graphics engine used. Figure 5 displays the test system, which was implemented using two different engines, an open-source java-based 3D engine named *javamonkey* (Fig. 5a) and a commercial Unity3D engine with the Oculus Rift HMD (Fig. 5b). Furthermore, using the distribution layer´s java interface, technologies such as Web Graphics Library (WebGL) can be integrated in order to enable the deployment of the software to a thin client architecture via web browser.

In the following paragraphs we describe the VR-based tool focusing on implementation aspects for managing the concurrent user access, as well as user interaction such as steering, object selection and manipulation.

3.2 Event-Driven Architecture for Managing User Access in 3DCVE’s

The networking architecture described in the previous paragraph offers robust, reliable and fast distribution of project-specific data amongst users participating in a joint planning project. Based on this architecture and using a data-centric model, visualization services were developed that enable the deployment of a multi-user VE via a client-server design. For this, the data-centric model describes the VE as collections of objects and their assigned attributes, such as translation, rotation etc. These collections are structured and synchronized among clients using the networking layer.

User actions are handled using an event-driven architecture, shown in Fig. 4. When a user evokes a change to an object in the scene using the UI (e.g. by rotating a 3D model), a change event is generated, encapsulated along with other changes on that object (e.g. pose) and sent to a command queue residing on a central server process. As described above, the server again resides on a separate JVM, which runs respectively on a Hazelcast node. The underlying distribution layer is also being used for the communication between clients and server through the change-command objects. The command

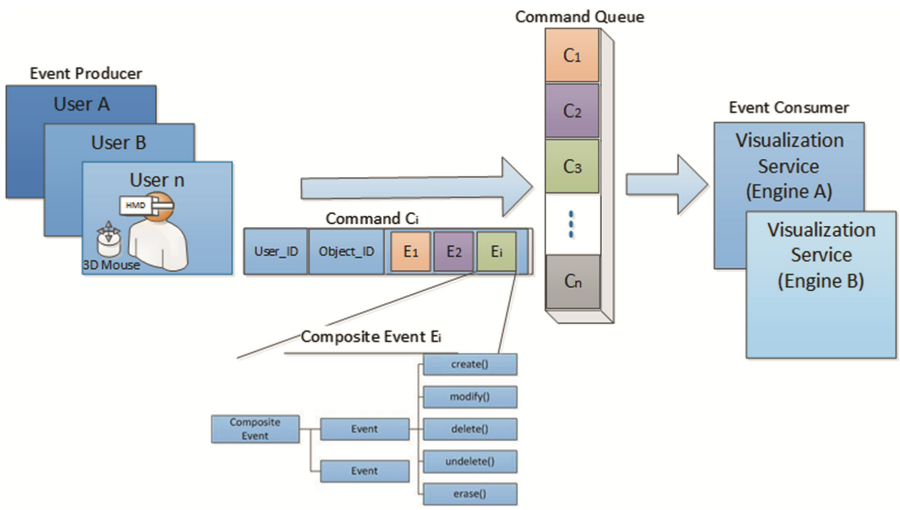

Fig. 4. Event-driven architecture for handling user actions (based on [17])

queue is executed serially processing every change, at which the server is authoritative over clients, being the only one that handles the requests that update the scene.

The above described architecture also sees for two important functions for a multi-user editor software, namely the creation of a full action history and the reversing of actions. Using the stored tagged events, a complete history of the actions of every user can be generated. This is particularly useful for documenting the design process and tracing back the users' decisions.

Another function, common to single-user software, is the undo/redo function, which offers the option to recover from errors and mishandlings. For a multiuser collaboration environment, the role of this function is even greater, since one user's errors can have a large impact on the actions of his collaborators. It is therefore important that errors can be quickly identified and reversed. The approach implemented in this work takes into account the model for a multiuser undo/redo function presented in [21], while modifying the concept to take advantage of the benefits that the data-centric distribution technology of KoDeMat platform offers. This was done using an additional, per-object buffer, which holds a history of changes performed on this object, as well as information about the user that performed the change. Applying a roles model that assigns rights to individual users or user groups, it is possible to manage the privileges of each user to "do" as well as to "undo" actions.

The concept also provides for the case of non-human triggered scene updates, namely for plain visualization of real and emulated MHS. In this case, there are highly volatile processes sending a large number of changes (e.g. position updates), a history of these changes in not desired and not meaningful. Each MHS needs only to provide status information about its own state into the VE and its agnostic of the changes that users perform. These machine clients communicate their data using the networking layer.

Managing Concurrent User Interaction. A challenge arising with simultaneous interaction of multiple users in a VE is concurrency control. According to Hudson [22], "a concurrency-control protocol ensures that incorrect behavior cannot occur as a result of concurrent access by multiple clients". In the area of computer-supported cooperative work, there is a number of approaches to address this issue. Some of the most popular involve *pessimistic approaches* that explicitly lock the objects that are being edited preventing its concurrent editing. Although this method is effective, it can lead to an inferior user experience as well as to diminishing collaboration effects.

The approach in this work takes advantage of the thread-safe concurrency mechanisms offered from the NoSQL, key-value store used for networking and the event-driven architecture discussed in the previous paragraph. There are no object-specific locks, that prevent multiple users to edit an object. Thus all changes (coming from an authorized user) on an object are being processed. In this direction, it is important for the distributed system to provide functions which raise user awareness and promote collaboration. One such function supports annotation for objects that are being edited by users. Figure 5b shows an object (robot) currently edited by user "fml_VR", thus signaling its status to others.

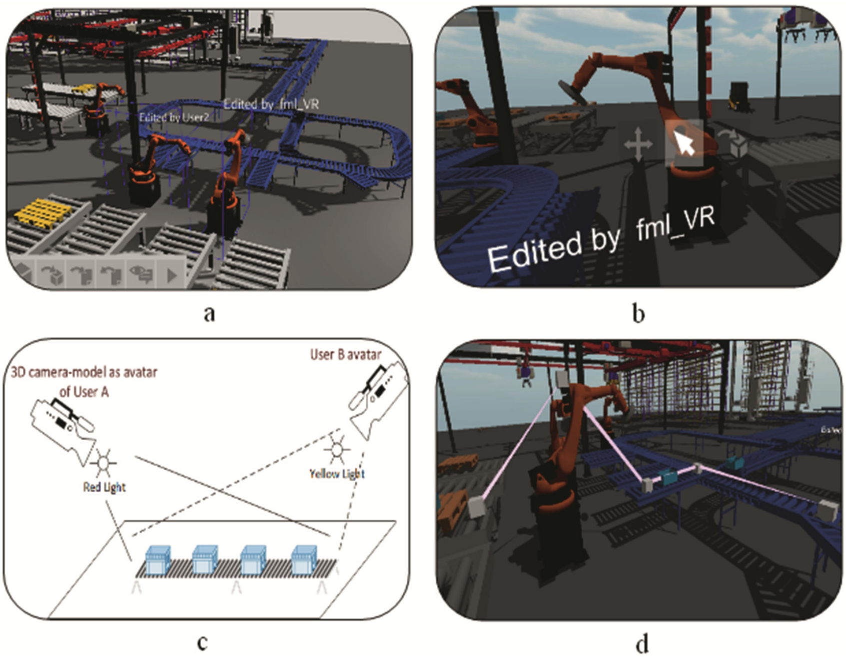

Fig. 5. Implemented 3D Collaborative Virtual Environment. (a): 3D Application with jmonkey 3D engine, (b) In-world menu for manipulating objects in Unity 3D, (c) camera models used as avatars, (d) definition of adjacencies between MHS in the VE

Orientation, Navigation and Object Manipulation in the 3D Space. As previously stated, this VR-enabled software uses the Unity 3D Engine and Oculus Rift HMD. In this VE, users can synchronously view and edit a 3D layout of the MHS as well as the surrounding facility infrastructure.

Egocentric orientation was used with a gaze-directed type of steering. Users can navigate in the 3D space, using a fly-through type of navigation. Through the integration of a 6 DOF controller in form of a 3D-mouse, and the head-tracking capabilities of the HMD, a 3D location-pointing concept was developed using a pointing arrow for the navigation in the 3D space to denote forward direction.

Various editing functions are implemented in order to enable the collaborative editing of a facility layout. Through an in-world 3D menu, users can perform actions such as moving, rotating and deleting of objects. The menu is context-based, displaying semi-transparent icons that depend on the selected object. Using a combination of raycasting and head-tracking techniques, users can select the desired function to execute, as shown in Fig. 5b. Further functions enable users to specify the adjacencies between MHS, by defining edges and vertexes of the material flow network (Fig. 5d). This way, alternative material flow matrices can be generated that can later be validated and used for the analysis of material flow.

Another important aspect in collaborative VE is the one of virtual presence. For the purposes of our facility planning tool, we chose to virtually represent users through non-human avatars. This approach represents users through a camera 3D model as shown in Fig. 5c. The benefits of such camera-avatars as collaboration tools are twofold. On the one hand, the avatar suggests the position of a user, on the other hand it is used to show other users in the collaborative VE a user's viewpoint. For this purpose, we extended the concept by using a type of telepointing approach. As a means to facilitate communication in the VE by attaching a light to the users' avatar, the viewing field of one user is being highlighted and so other users can see where someone is looking at.

4 Conclusion

In this work we presented a distributed 3DVCE, which can be used for collaborative planning of heterogeneous MHS's in logistics facilities. The paper's scope demonstrated how users can edit a 3D layout of a logistics facility and define adjacencies between systems. Further functions, such as specified communication interfaces between systems require editing digital documents, which is not as yet intuitive in a 3D space. It should also be noted that audio support communication was not implemented in the system. For this, it is assumed that a form of oral communication, for instance through a call conference, is available.

Advances in desktop-VR technologies with the introduction of commercial VR systems (the majority in the form of HMD) have made VR more accessible without requiring the large and expensive infrastructure of CAVE systems. This makes it possible to integrate VR technology in the planning processes of factories and logistics facilities. The introduction of 3DCVE's in planning processes can greatly assist in the early identification of misconceptions and uncover design-related production or implementation problems, while it contributes to more efficient and transparent processes and results in a significant reduction of costs and time (-to-market).

Nevertheless, a number of technical and non-technical challenges have been identified which need to be addressed in further research. The most important of the non-technical issues involves the integration of VR-technologies in existing planning workflows. The usability of 3DCVE should be extended by including more planning-specific tasks that should be identified based on concrete planning scenarios. Regarding technical challenges, immersive, VR-based tools and non-immersive (e.g. desktop-based) applications' interactions present still a major challenge. In order to achieve a truly cross-platform user planning interaction, the interplay between such systems should be enhanced to a degree where users can employ multiple platforms to achieve the same design tasks.

References

1. Kagermann, H., Wahlster, W., Helbig, J.: Recommendations for implementing the strategic initiative INDUSTRIE 4.0. Acatech, Frankfurt am Main (2013)
2. Mayer, S. H.: Development of a completely decentralized control system for modular continuous conveyor systems. Dissertation, University of Karlsruhe (TH) (2009)

3. Kipouridis, O., Roidl, M., Günthner, W.A., ten Hompel, M.: Cloud-based platform for collaborative design of decentralized controlled material flow systems in facility logistics. In: Kotzab, H., Pannek, J., Thoben, K.-L. (eds.) Dynamics in Logistics. Lecture Notes in Logistics, pp. 313–322. Springer, Heidelberg (2015)
4. Bullinger, H.-J., Potinecke, T., Tippmann, V., Rogowski, T.: Virtual engineering. Challenges and solutions. In: Proceedings of the 11th International HCI, vol. 5, p. 531 (2005)
5. Cecil, J., Kanchanapiboon, A.: Virtual engineering approaches in product and process design. Int. J. Adv. Manuf. Technol. **31**(9), 846–856 (2007). Springer
6. Siller, H.R., Vila, C., Estruch, A., Abellán, V., Romero, F.: Managing collaborative process planning activities through extended enterprise. In: Wang, L., Nee, L., Yeh Ching, A. (eds.) Collaborative Design and Planning for Digital Manufacturing, p. 153. Springer, Heidelberg (2009)
7. Bracht, U., Geckler, D., Wenzel, S.: Digitale Fabrik, Methoden und Praxisbeispiele, VDI. Springer, Heidelberg (2011)
8. Gausemeier, J., Rammig, F.J., Schäfer, W. (eds.): Design Methodology for Intelligent Technical Systems. Lecture Notes in Mechanical Engineering. pp. 12–19. Springer, Heidelberg (2014)
9. Grundig, C.: Fabrikplanung, Planungssystematik - Methoden - Anwendungen, pp. 37–40, Hanser, Munich (2014)
10. Frécon, E., Nöu, A.: Building distributed virtual environments to support collaborative work. In: Proceedings of the ACM Symposium on Virtual Reality Software and Technology, pp. 105–113 (1998)
11. Oliviera, J., Shen, X., Georganas, N.: Collaborative virtual environment for industrial training and e-commerce. IEEE VRTS (2000)
12. Wright, T., Madey, G.: Survey of Collaborative Virtual Environment Technologies. University of Notre Dame, USA (2008)
13. Pick, S., et al.: A 3D collaborative virtual environment to integrate immersive virtual reality into factory planning processes. In: 2014 International Workshop on Collaborative Virtual Environments (3DCVE) (2014)
14. Weidig, C., et al.: Future internet-based collaboration in factory planning. Acta Polytech. Hung. **11**(7), 157–177 (2014)
15. Menck, N., et al.: Collaborative factory planning in the virtual reality. In: 45th CIRP Conference on Manufacturing Systems (2012)
16. Mistrik, I., Grundy, J., van der Hoek, A., Whitehead, J.: Collaborative software engineering: challenges and prospects. In: Mistrik, I., Grundy, J., van der Hoek, A., Whitehead, J. (eds.) Collaborative Software Engineering, pp. 389–403. Springer, Heidelberg (2010)
17. ten Hompel, M., Günthner, W.A., Roidl, M., Kipouridis, O.: KoDeMat - Befähigung von KMU zur kollaborativen Planung und Entwicklung heterogener, dezentral gesteuerter Materialflusssysteme, Report, Lehrstuhl für Fördertechnik Materialfluß Logistik (fml) Technische Universität München - Lehrstuhl für Förder- und Lagerwesen (FLW) Technische Universität Dortmund, BVL (2014)
18. Sonnenfroh, M., et al.: Massively multiuser virtual environments using object based sharing. Electronic Communications of the EASST, vol. 17 (2009)
19. Veentjer, P.: Mastering Hazelcast, the Ultimate Hazelcast Book, Hazelcast Inc. (2015). http://www.hazelcast.com
20. Evans B.: An Architect's View of Hazelcast, White Paper, Hazelcast Inc. (2015). http://www.hazelcast.com

21. Choudhary, D.: A general multi-user undo/redo model. In: Proceedings of the Fourth European Conference on Computer-Supported Cooperative Work, ECSCW 1995 (1996)
22. Hudson, T.: Concurrency control for collaborative 3D graphics applications. Technical report, University of North Carolina at Chapel Hill, US (2000)