www.cadanda.com Computer-Aided Design AND APPLICATIONS

Integration of Virtual Reality in a Knowledge-based Engineering System for Preliminary Configuration and Quotation of Assembly Lines

Marco Rossoni[1] , Luca Bergonzi[2] , Giorgio Colombo[3]

[1]Politecnico di Milano, marco.rossoni@polimi.it
[2]Politecnico di Milano, luca.bergonzi@polimi.it
[3]Politecnico di Milano, giorgio.colombo@polimi.it

Corresponding author: Marco Rossoni, marco.rossoni@polimi.it

Abstract. Realistic visualization of products is now a must-have for all companies facing worldwide and highly competitive market. Despite Virtual Reality technologies are appealing, its industrial use is still limited to conceptual design and prototyping activities. One of the reason is that generating Virtual Reality (VR) environment is a complex and time-consuming task, especially for complex products or systems. Many technical data are involved in their design and configuration. A meaningful example is the preliminary configuration of assembly lines devoted to deliver a quotation to the customer. To be competitive, the quotation should be completed in tight time and contain variants of the configured system ranging different costs. Moreover, high-impact and successful quotation goes beyond the merely technical aspect. In this view, the automatic generation of a virtual reality environment can foster the adoption of this technology in industry, since its setup time is short and doesn't require any skills. In this paper, the integration of a VR module in product configuration and quotation process is proposed. The framework is a Knowledge-based Engineering (KBE) system that, taken the customer requirements as input is able to automatically generate a bunch of different solutions. Starting from technical data coming from a KBE system, a virtual environment is generated automatically fitting the features of the configured solution. Furthermore, the immersivity of the VR scene is enhanced by integrating the animation of the objects, like robots and pallets. After a brief description of the KBE system, the paper details the information is involved in, the implementation of the VR module and its integration within the KBE framework.

Keywords: Virtual Reality, Assembly Line, Quotation, Knowledge-based Engineering
DOI: https://doi.org/10.14733/cadaps.2019.329-344

1 INTRODUCTION

The worldwide market forces companies to deliver high quality quotation services to their potential customer. In fact, they should provide a range of different offers respecting tight lead-time. If the product to be marketed belongs to the so-called category of the "complex systems", the bidding process may be challenging. The manufacturing plants are part of that class: their design involves many activities and results in the physical layout of the plant. In the last years, researchers put an increasing effort on Information and Communication Technology (ICT) tools for automatically designing, optimizing and giving quotes of manufacturing plant layout [2]. Knowledge based Engineering (KBE) oriented to automatic configuration of products/systems is a powerful technology can tackle those issues [13]. In literature, many studies treat about advantages of KBE approach and its integration within several industrial environments. Usually, those applications deal with the automatic generation of technical documentation (reports, Computer Aided Design models, simulation outputs, and so on) [14, 6]. Nowadays, high bidding success rates go beyond the merely technical aspect.

Computer Aided Design (CAD) representation of technical solutions is a must-have for all bid annexes: it aims at providing documentary evidence of the final product/system architecture. Analyzing the role of technical drawing for project documentation, the authors notice a trend that put the non-expert users in center: from 2D drawings by hand, to digital 2D drawings, and finally to 3D model. The user, especially the non-expert one, can better understand the proposed solution with a 3D representation, facilitating also revisions and the feedback gathering process. A step further in this direction could be the model visualization using VR technologies. Nowadays, parametric, feature-based 3D CAD modeling is the state of the art for all those companies operating in industrial sector. A lot of research, both from academia and industry, has been lead for integrating CAD tools within Product Lifecycle Management (PLM) tools. Even if, this integration has not been completed yet, a good level of robustness and reliability have been reached: the proof is the wide adoption of PLM and CAD tool in everyday design activity. As mean of representation of technical data, 3D CAD models are more oriented to expert technicians rather than generic audience. From a marketing perspective, this aspect is not a good point: a proof could be the always greater use of static render of the product often included in quotations. Despite the wide use of CAD tools for everyday technical activities, some drawbacks appear evident from a marketing perspective. The authors identify three main flaws. The first is the difficult navigation through the 3D model of non-expert users. On the same wavelength, the second reason is the lack of stable interface with advanced displays (e.g. HMD) provided by mainstream CAD tools: just a limited number of devices are supported and the walkthrough is arduous. This is also reflecting back on hard collaborative working sessions. The third is referred to the limitation given by CAD data exchange file format. In order to ensure interoperability among different users, proprietary CAD files are usually translated in neutral file format (e.g. STEP, IGES, and so on). Despite many improvements have been done for making this translation lossless [8], some information is missed during this step. For example, standard way to create an assembly is by using mates: any neutral file format provides for this specification and the final 3D model results in a "simple" description of geometries and relative position. Other issues regard realistic textures of the surface as well as kinematic relation among parts. Some of the aforementioned issues could be mitigated through the adoption of VR representation. Few attempts have been done for integrating VR in PLM. Stelzer [12] proposes a possible approach to promote the use of VR system as the technical base for an internal or external collaboration platform. The proposed solution will make possible not only to edit and display shapes but also to integrate kinematic capabilities and depict non-geometrical product information, taking advantage of PLM system. Automotive domain already exploits of VR technologies at different tiers of product lifecycle, as reported in [7]. Mock-up and conceptual design, virtual prototyping, virtual manufacturing, virtual assembly and training activities are well explored areas where VR can effectively make the difference. Realistic visualizations (e.g. Virtual Reality) may increase success rate of the offer and improve the customer experience for any kind of product [1]. Many companies make use of Virtual Reality (VR) in everyday product design process. For example, Volkswagen has adopted VR for product design activities since 2008 [16], New

Holland VR Laboratories employ immersive environment for further ergonomic evaluations such as reachability of handles within vehicle buck [9]. All those applications concern product design process enhancements. Also in academia, many studies focus on effectiveness of this technology in various design activities, like training for maintenance [4], assembly tasks [5] as well as in fashion industry [15].

Beyond the state of the art, the scope of the paper is to promote the use of VR for the preliminary configuration at the commissioning stage: a virtual demonstration of the assembly line will be shown to the potential customer during the quotation submission. It doesn't replace standard CAD or PLM tools but it works beside them. This will be an addition to all the technical and commercial documents which as to be presented during the quotation. VR demonstration of the assembly line would enhance the customer confidence on the proposed solution and also helps in enhancing the chance to acquire the bid. The integration of the VR module within the KBE system represents an added value of the proposed solution, exploiting technical data to automatically build an accurate and detailed virtual environment. This paper aims at investigating the integration of virtual environments in a typical scenario of system configuration made by KBE tools. The focus is on automatic (i.e. without or limited human intervention) setup of VR scene starting from technical data to enhance engagement of customers through realistic walkthrough of plant building. The proposed approach is intended to encourage first-time-right configuration, reduce time for setup virtual environment and boost knowledge reuse up.

The paper is organized as follow: a description of the activities related to the quotation process as well as the framework and general architecture of the KBE system are depicted. Then, the generation of virtual environment based on standard libraries and its integration within KBE framework is presented. The proposed approach is applied to a case study regarding an assembly line layout visualization.

2 KBE SYSTEM FOR PRELIMINARY CONFIGURATION OF ASSEMBLY LINES

A software prototype has been developed using KBE approach, then tested on a case study proposed by an industrial company leader in automated manufacturing systems. The application deals with the automatic configuration of assembly lines for automotive industry. More in detail, the frame is the bidding process. A customer asks the proposal team of the company to set a price for a specific product/system. The goal is to provide to the customer a quotation that is as close as possible to the final product/system in a very limited time. This aim translates in providing a bunch of feasible project alternatives, ranging different bundles of cost, with a lead time as short as possible and with a high level of details. The configuration process takes advantage of pre-designed items and aims at establishing the layout to reach desired performance. In a nutshell, the structure of the product/system is defined as follow: pick pre-defined module (archetypes), arrange them in a feasible configuration, make use of aggregated parameters to compute performance of the configuration, estimate the cost and, finally generate the documentation. In this framework, a KBE system is useful to automate repetitive tasks and implement best-practices (or enhance them). Furthermore, it allows to move information that usually is employed in detailed phase in the preliminary one fostering a more accurate estimation of the performance and, consequently, "first-time-right" layout design. The next sections aim at describing the KBE system and its functionalities for better understanding of the rationale behind this work. All the examples are referred to the use case (i.e. assembly line domain).

2.1 Overall Process

The description of the overall configuration process is addressed in this section: Fig. 1 shows the macro-steps performed to configure an assembly line through an IDEF0 schema. Three different modules in the KBE system correspond to the main three steps in which the configuration process has been spit off. Fig. 3 details the "Visualization and Reporting" phase, that is the focus of the paper.

The starting point of the quotation process corresponds, from a technical point of view, to the preliminary

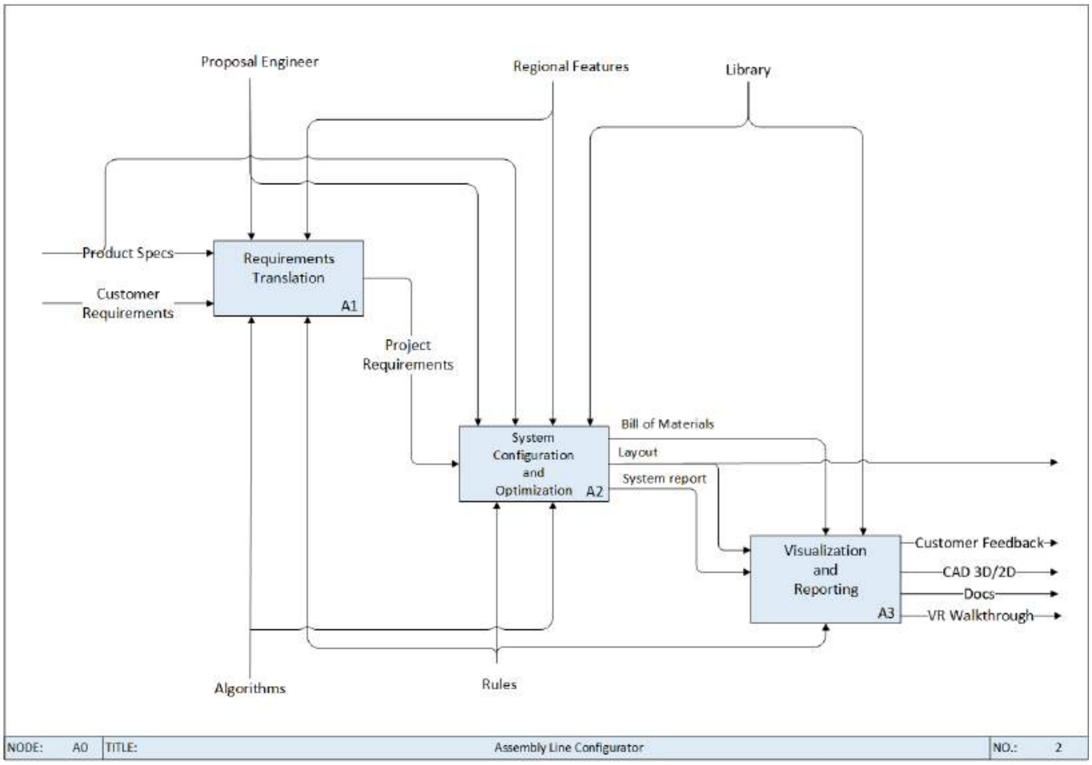


Figure 1: Schematic representation of overall process.

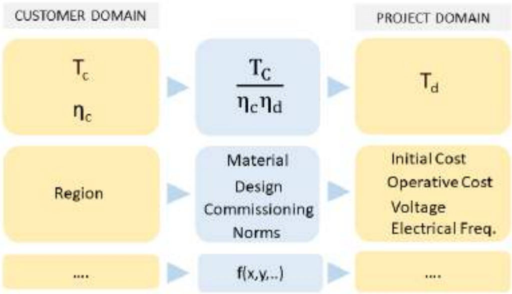


Figure 2: Examples of translation of customer requirements in project requirements

configuration process. The input is the Request for Quotation (RFQ) provided by the customer: this document include a set of specs the future system must accomplish as well as deadlines to be esteem. At this point, the information is usually unstructured and they don't follow any specific template. Hence, the proposal team has to scan the content of the RFQ and transform customer requirements to project requirements. The KBE system carries this task out automatically exploiting rules and algorithms developed ad-hoc for this scope. Best-practices are also implemented within those algorithms and rules. A meaningful example is the translation of required throughput (T_c) and efficiency (η_c) in design throughput (T_d) suitable for the configuration of the system. The customer specifies the number of item the plant must deliver each year (T_c) and the acceptable technical efficiency η_c. The company does not apply directly those values to design the plant: T_d is taken as reference value for the configuration process (Fig. 2). Its computation is based on a company best practice and the formula is shown in Fig. 2: to be conservative, T_d is increased by both technical efficiency (η_c) and design efficiency (η_d). The value of η_d is usually set at 0.85, hence T_d is overrated by 15%. Another meaningful example regards regional parameters. The customer specifies the target region for the system within the RFQ (i.e. the region/country/state where the plant will be installed). Material, design and commissioning cost as well as normative for the specific geographic area are exploited to extract the basic initial and operative costs as well as design specifications for each item will considered for the configuration (Fig. 2).

The output of this module are a set of structured information needed for designing and optimizing the configuration. The list of those information includes (but it is not limited to) target performances, product specifications (general sizes and weight of what has to be assembled), sorted list of tasks to be accomplished during the assembly process, and for each this task, a set of technological solution (module) able to perform it, featured of costs, failure parameters, execution time. A technological solution represents an aggregation of different physical objects able to perform the specific task (see Fig. 5): for a workstation, they should be the structure, the item devoted to be in contact with the workpiece (i.e. the end effector of a robot or the toolkit for human operator), the feeding system (i.e. items that supply raw material to the workstation) as well as the system devoted to perform the movement (i.e. the robot itself for automatic station or the human

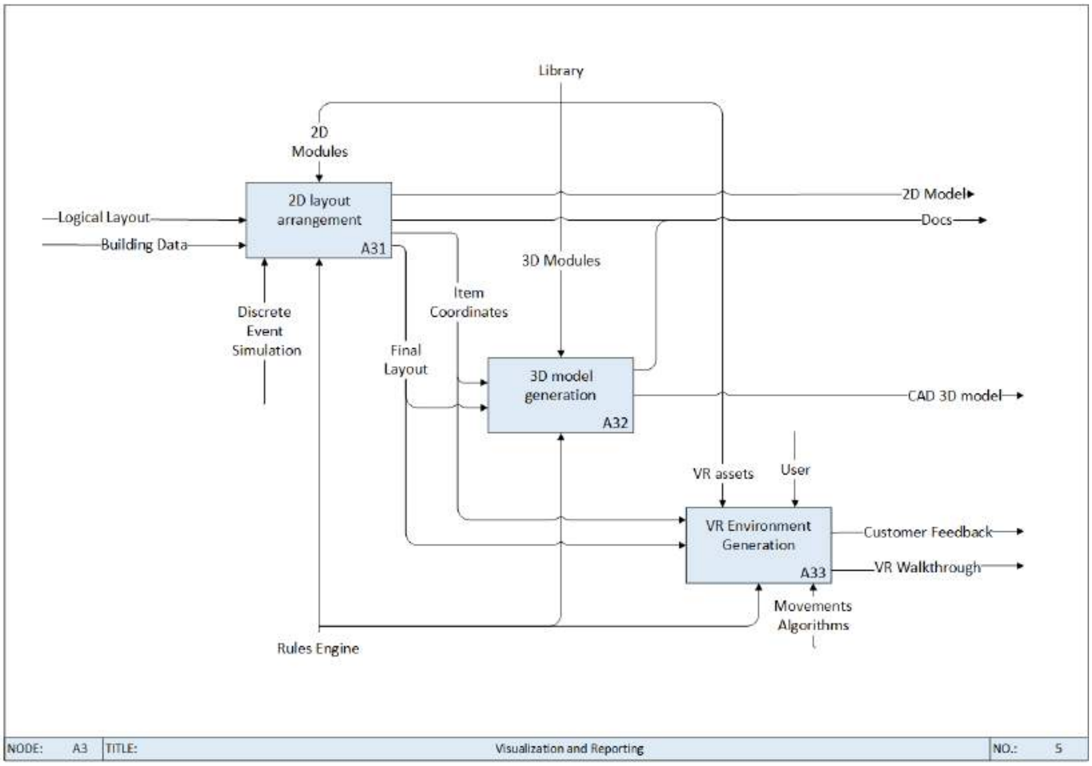


Figure 3: Schematic representation of Visualization and Reporting.

operator for the manual ones).

In this way, "System design and optimization" module is able to perform computation in order to select the optimal arrangement and combination of module among the available ones. This module performs the so called "logical design": assembly line balancing and buffer allocation are performed. Different methodologies are available to perform the aforementioned activities but their description go beyond the scope of the paper. Please refer to [11] for further details on how this module has been developed and integrated within the KBE system under description. The output of this module is the layout in terms of number and type of workstations, task(s) assigned to each workstation, number of pallets per buffer and the overall cost of the line, both initial and operational costs. Due to the complexity of the computation (i.e. assembly line design is well known as an NP-hard problem), physical constraints are not taken onto consideration in this phase.

Grounding on the results of "system design and optimization" module, the KBE system provides a user-friendly and powerful way to complete the configuration process: "visualization and reporting" module does take care of translating "logical layout" into a concrete one and generate the required documentation for the project. This module works in a "supervised" way: the user interaction is requested. Next section describes this module in detail.

2.2 Visualization and Reporting

The last block of the configuration process deals with finalizing the configuration and generate the proper documentation to submit to the customer. A detail zoom of "visualization and reporting" node is shown in Fig. 3. The "2D layout arrangement" is needed for finalizing the configuration process. As explained in Sec. 2.1, the "system design and optimization" is based on a mathematical representation (i.e. a model) and does not take care of all the variables needed for completely define the configuration. The first task is put in the building space all the workstation previously computed while respecting some additional constraints: the final layout must fit the building area, it doesn't overlap with unavailable areas (e.g. pillars, logistical areas, and so on) and the distance between two workstations must respect the minimum imposed by the "number of pallets per buffer" previously computed. A user friendly and step-by-step procedure has been implemented: each block, both workstations and buffers, can be placed within the building area thanks to a simple drag and drop. Starting from the first item of the assembly line (usually, a workstation), the user drags and drops each item in the workplace: only suitable items are proposed to the user that is forced to choose among them. This step-by-step procedure drives the configuration towards a feasible layout. The shape of the layout and the distance among workstation impact on the performance of the line: a discrete event simulation has been embedded in this module in order to check if the final layout meet the target performance. The final layout results in a 2D representation of the line. The conveyor system path (i.e. the path which the pallets are constrained to) is also discretized in a set of 2D coordinates: they will be useful to animate the pallets along the assembly line as detailed in Sec. 3.1.3.

The 2D coordinates of each item and its features are retrieved from the final layout while the respective 3D parametric representations are retrieved from a database (i.e. library). The 3D models are adapted to the scope of the configuration: for example, the height of the conveyor system is one of the parameter and it is determined by taking into consideration the type of the workstations (i.e. with human operator or fully robotic) and the region in which the plant are going to be installed. Then, the generation of the 3D CAD model is just a matching between the coordinates of each item and its 3D representation. The result is the 3D model of the assembly line. The last step is the automatic generation of a dynamic VR environment: the scope of this paper is to describe the integration of this environment in a technical solution and starting from technical data. Usually, VR environment is not delivered as final output of a configuration because they setup time is usually high. The added value of the approach proposed in this paper is the automatic generation of this kind of visualization in an automatic way. The rest of the paper is devoted to the description of these step and its integration within the KBE system described so far.

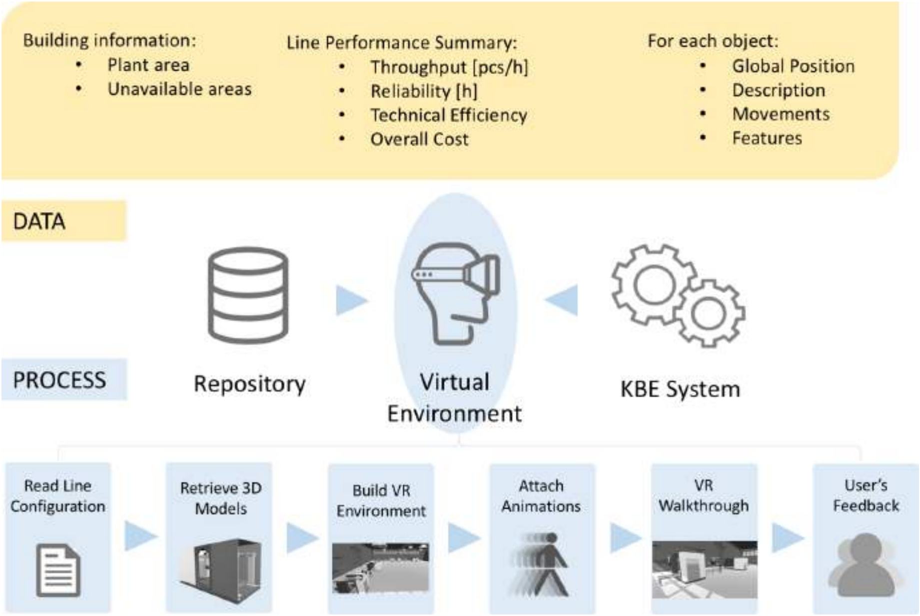


Figure 4: Data and information exchange between repository and VR module.

3 INTEGRATION AND DEVELOPMENT OF VIRTUAL REALITY MODULE

The Virtual Reality module concerns the implementation and the integration of a VR environment within the framework previously presented, coupling a design and simulation system with realistic visualization and interaction. The internal structure and main functions of this module are those in the A33 block of the IDEF0 representation of Visualization and Reporting (Fig. 3). This section discusses the development and deployment of the VR environment coupling the visualization system with a design and simulation system, described in Fig. 1 as the KBE system. Focusing on input required by this module and its integration with the model repository, main advantages combining detailed information and immersive environment have been outlined. The generation of the VR scene has been set up in such a way to be built in a semi-automatic way, according to data collected by KBE system. Currently, VR model preparation phase requires manual adjustments while the phase of static and dynamic environment generation has been designed to be completely automatic. In order to guarantee the most realistic visualization, also object animations have been included in this application and they are computed real-time by the software.

Fig. 4 shows how data, process and information are gathered and managed within the developed application. KBE system plays the role of dynamically connecting technical data with visualization module. On the top level, data regarding building information, line performance, and detailed information for each selected object have been considered for the VR environment generation. The VR module also requires compatible 3D representations of objects composing the assembly plants, currently available only in native and, usually proprietary, 3D CAD file format. Starting from the line configuration provided by KBE system, this application retrieves 3D models from the repository and then builds the environment, correctly placing objects (e.g. workstations, conveyors) within the available building area. Finally, a virtual walkthrough at the plant is run and users' feedbacks are collected. All the project stakeholders can visualize and interact with the virtual environment through VR devices as Head Mounted Displays (HMDs). For multiple users, one person can wear the HMD, while others could watch the external display to understand what the HMD user is seeing.

Evaluating visual and functional issues, customer proposes some changes or finalizes the commissioning of the line. In this way VR can helps in reduction of cost, time, risk and improves quality and communication during commissioning of the line. Following sections will explain the preliminary activities to be done before running the VR walkthrough, how the VR scene is automatically generated and how the movements are integrated.

3.1 Preliminary Activities

Some activities have to be carried out before running the VR walkthrough. They aim at preparing the set of information needed for setup the VR scene and, translate them in a proper format if needed. Three main activities must be done: translating 3D CAD models in a proper format, retrieving the information about the configured solution and computing the movements to animate the VR environment.

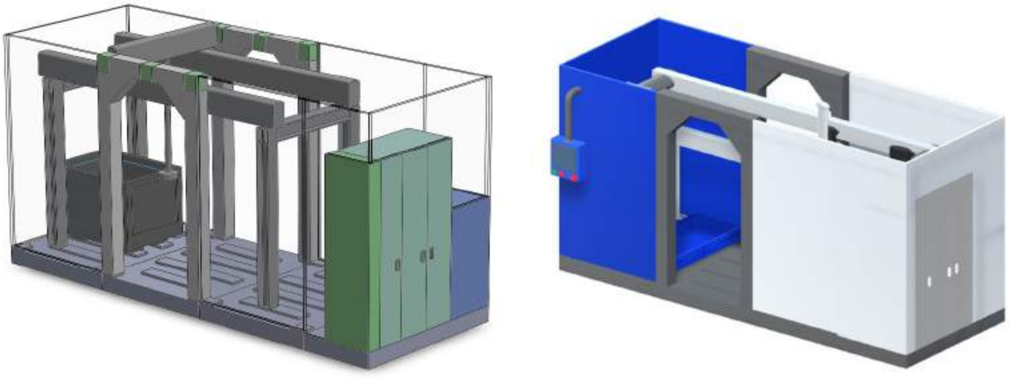
3.1.1 From 3D CAD to VR Assets

Usually, design and quotation of assembly plants requires technical and functional specifications and the entire bidding process ends with the realization of a 3D CAD model of the entire line. An expert usually pick the 3D models up from a repository (e.g. often Product Data Management or Product Lifecycle Management tools in industry) and arrange them in a layout that fits the requirements of the customer. As described in Sec. 2, the KBE system (partially) substitutes the expert: in this view, human gets role as supervisor of the process. The company repository already contains the whole library of 3D CAD models of workstations and the equipment that are necessary to build the virtual representation of the assembly line. However, virtual reality softwares have not been designed to directly accept as input the native CAD files: neither proprietary either neutral file format is accepted. Furthermore, CAD representations is mostly geometrical while the VR environment requires more visual details (e.g. detailed textures, materials, and so on).

The automatic generation of a virtual environment requires those models to be accessible as soon as they are needed. Two ways are available: the automatic conversion of CAD model at runtime to VR representation or a library holding ready-to-use VR representation beside the existing repositories. The former is not attainable since the lack of visual details: they are not retrievable from any sources and their application has to be done manually, at the moment. Furthermore, the runtime conversion may considerably slow down the building process despite it would ensure a high flexibility: adding an object to the repository would not involve any manual conversion. Hence, the library is an exact replica of the CAD models enriched with detailed visual details (Fig. 5). Pre-processing phase must be manually carried out, requiring a lot of time to convert and store data in the repository. Integrating the application with an automatic conversion of 3D CAD model into VR-ready models is a complex issue: for example, it requires automatic recognition of surfaces for applying the correct textures. In this paper, the authors followed the latter approach: a VR object library has been built. This application has been developed with Unity 3D, and only objects in FBX or OBJ format can be accepted. The conversion procedure generates objects with no colors, textures and material properties, being not ready to be visualized into VR environment. For this reason, the user must match parts with the correct aesthetic properties. Focusing on the phase in which textures must be added to create an immersive environment, in Fig. 5 from left to right, the transition from CAD to VR is shown.

Concerning the way in which VR objects will be placed in the scene, a global reference system for each one must be chosen accordingly to the placement procedure (Sec. 3.1). Respect to a 3D CAD software, VR engines are usually not based on the concept of "mates" for assembling complex products/systems, but the position is determined by spatial object coordinates and following a parent-child hierarchy. Summarizing, the steps for preparing the VR assets, for each item in the repository, are:

- convert the 3D model from CAD to VR suitable format (i.e. OBJ or FBX);
- apply materials and textures within the VR engine;



(a) CAD (b) VR

Figure 5: From 3D CAD to VR model.

- set the position of the reference system in the center of its bounding box;
- save the asset in the repository beside (i.e. referencing) the 3D models.

In this way, each item available for the configuration has its equivalent VR asset and the KBE system knows where the reference system of the object is.

3.1.2 Layout Information Retrieval

The set of information describing the layout of the specific configuration have been formalized using a metalanguage that is easily readable and understandable by users and software. Extensible Markup Language (XML) is considered one of the most suitable language for the definition and storage of data providing logical and hierarchic structure [10]. KBE system perfectly integrates XML for the preliminary design, and Fig. 6 reports an abstract of how the entire plant is described. Standard data formalization is fundamental to integrate different CAD and VR software, being able to make the approach more general and software-independent. In our approach, XML contains the information for generating 3D CAD model as well as the ones needed for generating the VR environment automatically.

Building specifications (e.g. plant area, pillars), pallet dimensions, workstations specs (e.g. model, equipment, position), conveyors sizes among the others are the information considered to build the virtual environment. On the basis of the specific assembly line, the XML file is generated by the "Logical Design" module (Fig. 1) and it is finalized after the "2D layout arrangements" step (Fig. 3). XML formalization bridges technical design with all the "Reporting and Visualization" activities.

3.1.3 Computation of Movements

Another set of information belongs to the movements to be assigned at the objects in the scene. The computation of animations is crucial to build a dynamic scene in such a way that the user can be involved in the VR environment. Concerning the animation of an assembly plant, robots and the pallet holding the assembly, need to be animated.

Fig. 7 shows the process to compute movements for robotic workstation. The inputs are the geometry of the robot and the trajectories of the end-effector. The former refers to the 3D CAD model retrieved from the repository. The trajectory are available in the repository of each pair robot-task: $[TR]_{ij}$ is the

```
▼<Project ID="1">
 ▼<Plant ID="1">
    <PlantArea Length="40" Width="50"/>
    <Pillars Number="0"/>
  ▼<Line ID="1">
     <Pallet Length="0.6" Width="0.6" ConveyorHeight="1000"/>
   ▼<Workstations Number="8">
    ▼<Workstation ID="1">
       <Model Name="Manual" Length="600" Benches="1"/>
       <Equipment Type=""/>
       <MinimumBufferLength Length="1"/>
       <Position XPos="6" YPos="3" Angle="0"/>
     ▼<Buffer ID="1">
        <Conveyor ID="1" Type="straight1" Length="1" PosX="6" PosY="5" Angle="0"/>
       </Buffer>
     </Workstation>
```

Figure 6: Assembly line XML structure.

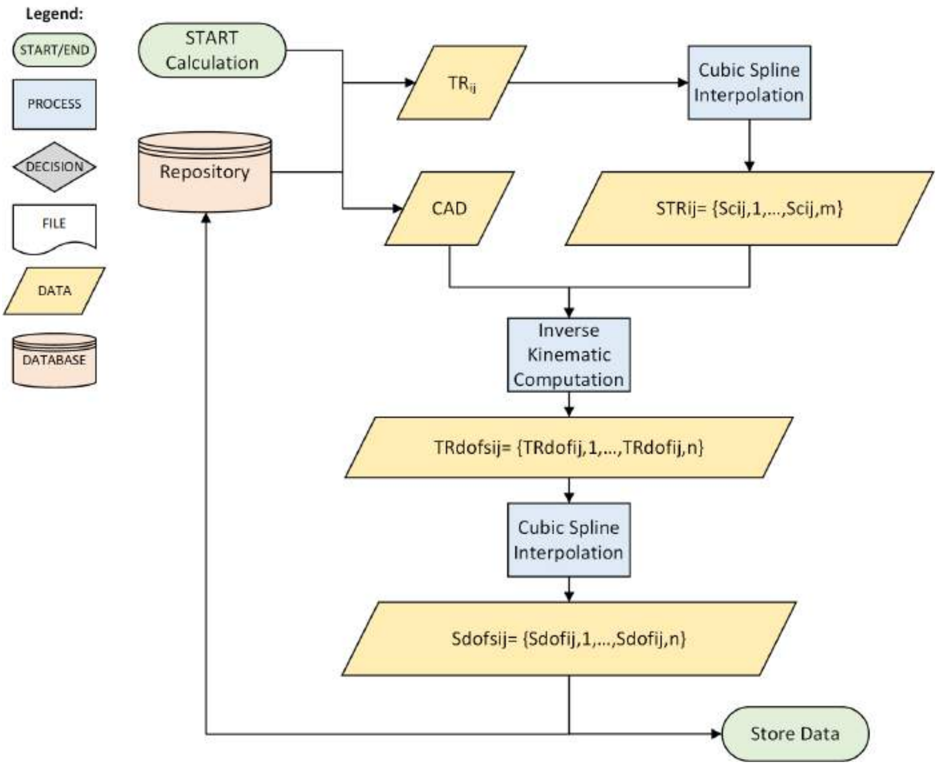


Figure 7: Computational process of robot animations.

end-effector trajectory matrix of i-th robot performing j-th task. $[TR]_{ij}$ has 6 columns that represent the set of 6 coordinates that describe the position and orientation of the end effector. The number of rows depend on the number of time step which the task duration is discretized in. $[TR]_{ij}$ is previously computed during the design of the line (block A2, Fig. 1). The goal (i.e. the output of the computation) is $[S_{DOFS}]_{ij}$: it contains the description of the trajectory for each Degree of Freedoms (DOFs) of the i-th robot performing j-th task in term of cubic spline coefficient. Each column of $[TR]_{ij}$ is first interpolated using a cubic spline interpolation: each direction/orientation is represented by the 6 coefficients of a spline, ensuring a smoothed path. Grounding on the spline representation and the geometry of the robot the inverse kinematic is applied. Then, the value for every degree of freedoms (DOFs) of the robot has been retrieved applying the inverse kinematic: it is performed exploiting an open source library, namely Pybullet [3]. $[TR_{DOFs}]_{ij}$ is the output of the inverse kinematic: it contains the trajectory of each DOF of the robot in term of discrete set of points. Also in this case, a cubic spline interpolation is applied. The final output is the $[S_{DOFs}]_{ij}$ that represent the trajectory of each DOF as a cubic spline function. This last step allows movements to be stored easily and de-couple the description of the movement from the refresh rate of the VR environment. Furthermore, in order to guarantee a fluid walkthrough and avoid possible lag at runtime, the computation of all animations has been done before starting the navigation within the VR scene. This loop is repeated whenever the coefficients for the pair robot-task (i.e. $[S_{DOFS}]_{ij}$) are not present in the database.

The computation of the pallets animation grounds on the set of 2D coordinates coming from "2d Layout Arrangement" block (Fig. 3) and the speed of the path determined during the design of the assembly line ("System Configuration and Optimization", Fig. 1). The path among the points is approximated as linear. In order to unburden computational cost at runtime, a lookup table is built (Eq. 1):

$$\begin{bmatrix} x_0 & y_0 & z & t_0 \\ \vdots & \vdots & \vdots & \vdots \\ x_i & y_i & z & t_i \\ \vdots & \vdots & \vdots & \vdots \\ x_n & y_n & z & t_n \end{bmatrix} \tag{1}$$

$$t_i = t_{i-1} + \frac{\sqrt{(x_i - x_{i-1})^2 + (y_i - y_{i-1})^2}}{V} \tag{2}$$

Where x_i, y_i are x and y i-th coordinates respectively, z is the height of the conveyor system, t_i is the i-th time computed as Eq. 2, n is the number of point which the path was discretized in, and V is the speed of the pallet. The summation of all t_i is the lead time of a pallet (i.e. the time from the loading of the pallet in the line and its unload), excluded the task processing time. VR engine (i.e. Unity 3D) does perform a linear interpolation of each pair (X_i, t_i), (y_i, t_i) and (z_i, t_i), deriving the animation curve to be attached to the pallet. In this case, cubic spline interpolation is not needed because each assembly line configuration has its own specific layout, hence a generalization of the approach does not make sense in the case of pallets movement and the authors prefer to abet computational performance.

The synchronization between the movement of the pallet and the movements of the robots will be described in Sec. 3.3.

3.2 Automatic Generation of Virtual Environment

Once all the preliminary activities have been completed, all the ingredients for the automatic generation of virtual environment are ready. To manage and automate the visualization process, Unity 3D has been chosen. Fig. 8 shows how the VR application works: it depicts all the computation performed at startup.

Given a specific configured solution, all the information are retrieved from the repository: to extract all the useful information to place objects and to assign task animations, the XML file (i.e. formal model) is loaded and parsed. Then, all the items constituting the configured solution are placed in the proper position

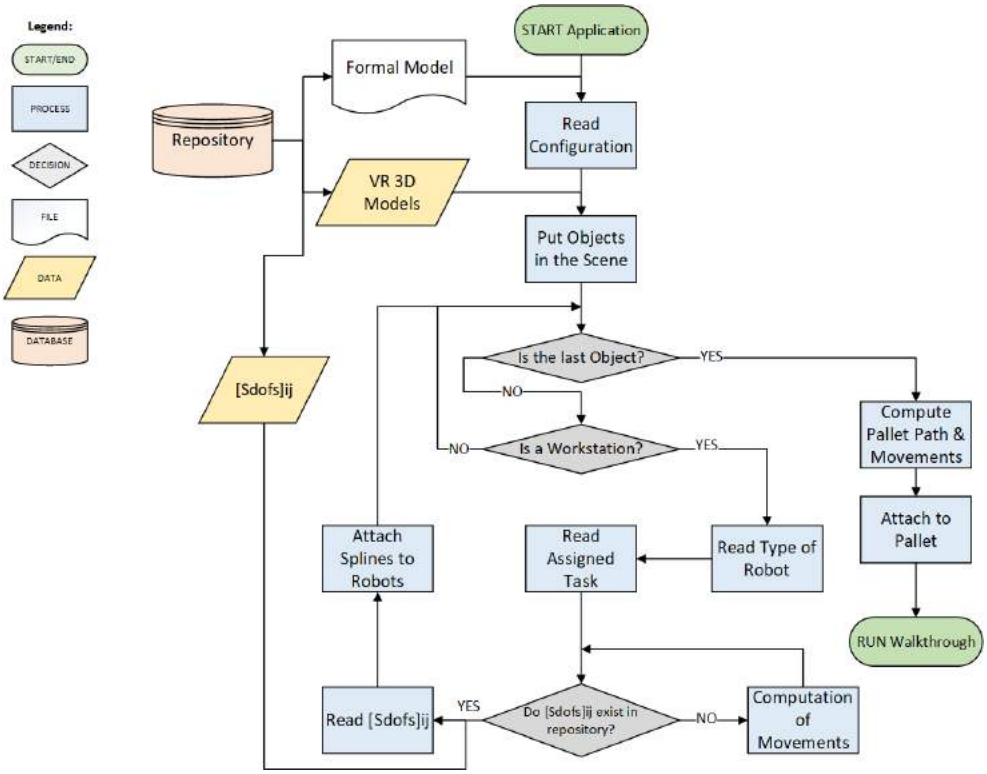


Figure 8: Flowchart of dynamic generation of virtual environment.

within the VR environment. For each workstation, the related animations are retrieved from the repository (as described in Sec. 3.1.3): if the related animations have not been computed yet, the calculation is performed making use of the process described in Fig 7. Once the movements for all the workstations are assigned, the lookup table (Eq. 1) is attached to the relative virtual objects.

The first-person controller is instantiated, and the active walkthrough of the plant is ready to be experienced.

3.3 Objects Animations Allocation

Mimic as much as possible the real behavior of what the virtual environment should represent is crucial. Referring to the design and configuration of an assembly plant, tasks carried out by each workstation become extremely important for both marketing and engineering evaluation activities. Integrating animations, main issues to face regards the dynamic association between each object and the task to simulate. Being the scene automatically generated every time the configuration changes, animations need to be calculated before the application starts, using a predefined library and tools (Sec. 3.1). In order to ensure a high level of fidelity, the animation of the objects within the scene must be synchronized. For example, a robot has to begin a task only when the pallet is in the correct position within the workstation. Hence, the integration among animations, objects, and Unity 3D has been deeply tuned in order to find an optimal solution in terms of computational power and VR environment effectiveness.

Fig. 9 shows how the data is fed in Unity 3D: integration of a 6 DOF robot is depicted. The animation curves previously computed is assigned to the respective subpart of the robot. The hierarchical structure allows each DOF spline to be assigned to correct subpart as well as to link the different part of the robot (like the mates do in CAD tools). Each DOF spline is evaluated by Unity 3D that interpolates over its own key frames to create a smooth animation.

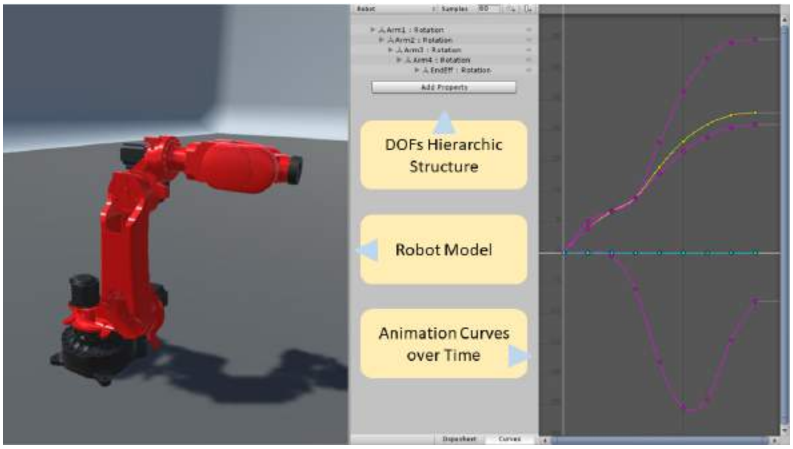


Figure 9: Spline interpolation of the DOFs trajectory in Unity 3D.

The synchronization of every animation has been done placing triggers along the pallet path. Assign a trigger just means raise an event (e.g. stop or begin a animation) when another event happen (i.e. two boxes collide). The triggers position corresponds to the center of each workstation: this is where the robot performs its activity. As the pallet passes through every workstation at a time, it stops and, the animation of the correspondent task is performed. Two additional triggers are put at the beginning and at the end of the conveyor system. Those control the number of pallet on the line (i.e. the Work In Progress) and tune the throughput of the line (i.e. units per minute). This strategy to synchronize the animation is trivial but very effective and flexible.

The next session describes the walkthrough and how the user can interact and leave feedbacks on the project.

4 WALKTHROUGH, INTERACTION AND FEEDBACK GATHERING

The visualization and interaction with the virtual environment are commonly considered the most important activities to evaluate and improve the design and development of complex systems or products. Focusing on the configuration of assembly plants, the VR scene allows the customer to be actively involved in the final part of the design process and to suggest adjustments. The enhanced visualization and the realistic perception of a product help the user to carry out a first-time right solution.

In Fig. 10, the VR environment generated with Unity 3D and the direct interaction with the configured assembly line are shown. Respect to 3D CAD visualization, the virtual reality guarantees a much higher effectiveness, simulating how and where the plant will be installed. Aesthetic details of building and workstations relies on 3D models prepared and stored in the repository, so they depend only on the level of detail with which they have been created. For this reason, the VR scene is always aesthetically improvable. The maximum immersion is provided by the integration of an HMD device to enable the active walkthrough around the scene. In this test case, Oculus Rift has been used. This device allows the user to look carefully at every technicality of the plant, being easier to judge potential defects of the designed configuration.

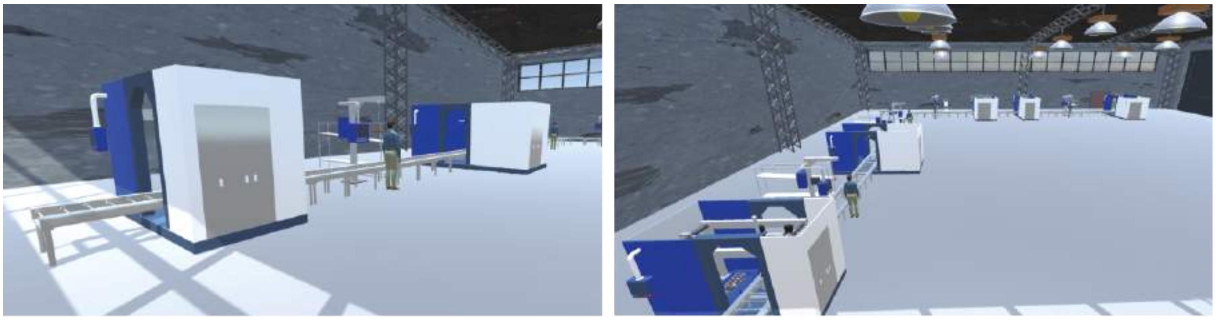


Figure 10: Two screenshots of virtualy reality scene.

Figure 11: Interaction with VR environment: Oculus Rift and display as visualization tool and a joystick as input device

The last step of an assembly plant configuration is the feedback gathering, allowing more intuitive and easy exchange of ideas between the designer and the customer. As Fig. 11 shows, the interaction with the virtual reality application has been provided by the use of and HMD device (i.e. Oculus Rift) and a joystick. The user can interact with the virtual environment moving the first person controller around the assembly plant giving first-look feedback of the proposed assembly line. In fact, this tool has been developed allowing the customer and the proposal engineer to see the same environment. The user can interact with every element of the VR scene, taking vocal notes addressed to specific objects. Assumptions and suggestions are then stored in repository: since the final solution still requires design iterations, they can be used to further development of the assembly line layout. This can be considered a new way to collect customer feedback in early design and configuration phases and it can reduce downtime due to possible misunderstandings between customer needs and who is in charge to build the assembly plant.

5 CONCLUSIONS

In recent years, design and configuration activities of complex systems, as manufacturing plants, have undergone many improvements due to the development of automatic generation tools for technical documentation and detailed visualization. In this paper, the integration of virtual reality visualization with a KBE system for the configuration and quotation of assembly plants has been presented. Taking into account layout information, a VR representation can be automatically built for the solutions previously configured by the KBE system, as long as the 3D models are available in the repository by manually converting them from CAD to VR-compatible format. The final goal of this approach is to provide an integrated tool offering a more immersive and intuitive environment, compared to current visualization tools (CAD), in which customers and designers can exchange ideas and feedbacks. Furthermore, it enables customer feedbacks gathering facilitating improvement of the configured solutions. In this phase, the main strategical advantage introduced by VR over standard CAD visualization is clearly visible: marketing activities will be enhanced following customer requirements and shortening time to make an appealing visualization.

The whole quotation process has been integrated with the KBE system that collect and manage all the customer requirements needed for the configuration of the assembly plant and the development of virtual reality module. Before setting up the environment, preliminary activities are required. Firstly, CAD models need to be converted into VR-compatible models, taking into account material textures and items reference systems. Then, the overall layout has to be retrieved using a formal model that collects all the information necessary to build the assembly plant. Finally, in order to make the VR simulation much realistic as possible, animations related to assembly tasks and pallet movements need to be computed. After preliminary activities, the automatic generation of the environment is done reading the formal model and placing the objects in the virtual scene. Grounding on movements previously calculated, animations have been embedded within the virtual world, paying attention to their synchronization. In the end, the walkthrough can be run and interaction with the surrounding environment is enabled. The user, wearing the VR HMD device, can interact with the assembly plant, observing the animation integrated with the whole environment and leave feedbacks in the form of vocal notes.

This approach can allow a great improvement of the configuration process considering the possibility to integrate the customer feedback into the further development of the assembly line layout. The combined use of new visualization technologies and automatic configuration processes can bring many advantages in early design phase reducing lead-times and downtime due to possible misunderstandings between customer needs and who is in charge to build the assembly plant, pro-actively involving the customer in such activity. From the marketing perspective, presenting at the customer a realistic visualization is more impressive rather that traditional CAD model.

ORCID

Marco Rossoni http://orcid.org/0000-0003-2714-9043
Luca Bergonzi http://orcid.org/0000-0003-0664-1098
Giorgio Colombo http://orcid.org/0000-0002-9999-8960

REFERENCES

[1] Barnes, S.: Understanding Virtual Reality in Marketing: Nature, Implications and Potential. SSRN Electronic Journal, 2016. http://doi.org/10.2139/ssrn.2909100.

[2] Colombo, G.; Furini, F.; Rossoni, M.: The Role of Knowledge Based Engineering in Product Configuration. In Advances on Mechanics, Design Engineering and Manufacturing, 1141–1148. Springer International Publishing, 2017. http://doi.org/10.1007/978-3-319-45781-9_114.

[3] Coumans, E.; Bai, Y.: PyBullet, a Python module for physics simulation for games, robotics and machine learning, 2017. http://pybullet.org.

[4] Ferrise, F.; Caruso, G.; Bordegoni, M.: Multimodal training and tele-assistance systems for the maintenance of industrial products. Virtual and Physical Prototyping, 8(2), 113–126, 2013. http://doi.org/10.1080/17452759.2013.798764.

[5] Gavish, N.; Gutiérrez, T.; Webel, S.; Rodríguez, J.; Peveri, M.; Bockholt, U.; Tecchia, F.: Evaluating virtual reality and augmented reality training for industrial maintenance and assembly tasks. Interactive Learning Environments, 23(6), 778–798, 2015. http://doi.org/10.1080/10494820.2013.815221.

[6] La Rocca, G.; van Tooren, M.J.L.: Knowledge-based engineering to support aircraft multidisciplinary design and optimization. Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering, 224(9), 1041–1055, 2010. http://doi.org/10.1243/09544100JAERO592.

[7] Lawson, G.; Salanitri, D.; Waterfield, B.: Future directions for the development of virtual reality within an automotive manufacturer. Applied Ergonomics, 53, 323–330, 2016. http://doi.org/10.1016/j.apergo.2015.06.024.

[8] Lorenz, M.; Spranger, M.; Riedel, T.; Pürzel, F.; Wittstock, V.; Klimant, P.: CAD to VR - A Methodology for the Automated Conversion of Kinematic CAD Models to Virtual Reality. Procedia CIRP, 41, 358–363, 2016. http://doi.org/10.1016/j.procir.2015.12.115.

[9] Monacelli, G.; Li, K.; Brokaw, A.: Virtual Reality Application in Case New Holland Product Development. In ASME 2011 World Conference on Innovative Virtual Reality, 217–223. ASME, 2011. http://doi.org/10.1115/WINVR2011-5543.

[10] Rezayat, M.: Knowledge-based product development using XML and KCs. Computer-Aided Design, 32(5-6), 299–309, 2000. http://doi.org/10.1016/S0010-4485(00)00013-0.

[11] Rossoni, M.; Colombo, G.; Furini, F.; Ascheri, A.E.; Ippolito, M.: Assembly Line Balancing and Configuration: An Alternative Approach for Design and Planning. In Volume 1B: 36th Computers and Information in Engineering Conference, V01BT02A007. ASME, 2016. http://doi.org/10.1115/DETC2016-60265.

[12] Stelzer, R.H.: Virtual Reality Based Engineering Collaboration as Part of the Product Lifecycle Management. In Volume 3: 30th Computers and Information in Engineering Conference, Parts A and B, 1327–1334. ASME, 2010. http://doi.org/10.1115/DETC2010-28250.

[13] Stjepandić, J.; Verhagen, W.J.C.; Liese, H.; Bermell-Garcia, P.: Knowledge-Based Engineering. In Concurrent Engineering in the 21st Century, 255–286. Springer International Publishing, Cham, 2015. http://doi.org/10.1007/978-3-319-13776-6_10.

[14] Tiwari, V.; Jain, P.K.; Tandon, P.: Design Decision Automation Support through Knowledge Template CAD Model. Computer-Aided Design and Applications, 12(1), 96–103, 2015. http://doi.org/10.1080/16864360.2014.949580.

[15] Vitali, A.; Rizzi, C.: A virtual environment to emulate tailor's work. Computer-Aided Design and Applications, 14(5), 671–679, 2017. http://doi.org/10.1080/16864360.2016.1273584.

[16] Zimmermann, P.: Virtual Reality Aided Design. A survey of the use of VR in automotive industry. In Product Engineering, 277–296. Springer Netherlands, Dordrecht, 2008. http://doi.org/10.1007/978-1-4020-8200-9_13.