Prototyping industrial workstation in the Metaverse: a Low Cost Automation assembly use case

Prototyping industrial workstation in the Metaverse: a Low Cost Automation assembly use case

1st Federico Manuri
DAUIN
Politecnico di Torino
Torino, Italy
federico.manuri@polito.it

2nd Nicola Gravina
DAUIN
Politecnico di Torino
Torino, Italy
nicola.gravina@studenti.polito.it

3rd Andrea Sanna
DAUIN
Politecnico di Torino
Torino, Italy
andrea.sanna@polito.it

4th Paolo Brizzi
Competence Industry Manufacturing 4.0 S.C.A R.L
Torino, Italy
paolo.brizzi@cim40.com

***Abstract*—Low-cost Automation (LCA) represents a relevant use case that can benefit from a design and prototyping step experienced in Immersive Virtual Reality (IVR). LCA is a technology that automates some activities using mostly standard automation components available off-the-shelf. However, since LCA systems should adapt to existing standard production lines and workstations, workers need to customize standard LCA templates. This adaptation and customization step is usually performed on the real, physical LCA system, thus, it can be very time-consuming, and in case of errors it may be necessary to rebuild many parts from scratch. This paper investigates the usage of an Immersive Virtual Environment (IVE) as a tool for rapid and easy prototyping of LCA solutions. The proposed system loads from a digital library the 3D models of the components and provides users a set of tools to speed up the LCA system creation in a virtual room experienced through an IVR Headset. When the user completes the creation of the LCA system, it is possible to simulate its physical properties using the Unity 3D Physical Engine. Moreover, it is possible to obtain a list of all the pieces needed to build the prototype and their dimensions, to easily reproduce them in the real world. To assess the usability of the proposed system, a LCA building task has been defined, whereas users had to build a LCA solution using a template model for reference. Results show that the system usability has been highly appreciated by both skilled users and inexperienced ones.**

***Index Terms*—Virtual Reality, Industry 4.0, Low Cost Automation, Virtual manufacturing**

I. Introduction

The advent of Industry 4.0 has boosted the usage of innovative technologies to promote the digital transformation of manufacturing companies. The nine main pillars of Industry 4.0 reflect the nine main innovations that the industry is encouraging and employing to enhance all aspects of the manufacturing process [1]. One of the key innovations that can successfully help the fourth revolution is represented by Virtual Reality (VR), more specifically exploiting the possibilities offered by cyber-physical systems and virtual environments (VEs). Digital Twins (DTs) have been widely adopted to virtually reproduce the physical world for training activities and simulations [2]. Moreover, virtually prototyping a complex workstation and/or an industrial production line can greatly reduce the design cost and time. Immersive Virtual Reality (IVR) can provide a whole new level of interactivity and easiness of use for designing and prototyping tasks [3]: IVR enables the user to virtually explore the environment, evaluating the efficacy of the design choices from different points of view and with different levels of detail depending on the availability of digital information pertaining to the real counterpart. Moreover, the ability to share IVEs with people far away connected through the Internet can enhance remote collaborations among users and can also be a way of circumventing the movement restrictions and isolation caused by the recent Covid-19 pandemic.

Low-cost Automation (LCA) represents a relevant use case that can benefit from a design and prototyping step experienced in IVR. LCA is a technology that automates some activities using mostly standard automation components available for off-the-shelf [4]. A LCA solution is generally characterized by: zero or very little power consumption, a few actuators, high flexibility, high reliability, small dimensions, minimum maintenance, and minimum investment/running costs. The Return on Investment (ROI) in terms of improved productivity and better work efficiency is high, as a LCA solution assures quality, provides flexibility, increases productivity, and reduces costs. The automation implemented by LCA systems is usually based on the concept of Karakuri Automation [5]: 'Karakuri' is a Japanese word, which means achieving motion with no power or low power. It is an ancient Japanese technique to move dolls without actuators or power. In the Karakuri technique, different mechanisms like rope and pulley, counterweights, cam and follower, etc., are used to achieve unique motion or to convert one form of motion into another (e.g., vertical to horizontal and linear to rotary). Combining these motions to form an automation solution requires an engineering approach. LCA solutions have been proven to be extremely useful to further reduce manual operations in highly automated systems and production lines [6]. However, since LCA systems should adapt to existing standard production lines and workstations, workers need to customize standard LCA templates. This adaptation and customization step is usually performed on the real, physical LCA system, thus it can be very time-consuming and in case of errors, it may be necessary to rebuild many parts from scratch. This paper is organized as follows: section 2 provides an overview of the state of the art whereas section 3 depicts the paper contribution. Section 4 describes in detail the proposed system and the functionalities deployed to help the user in LCA assembly tasks. Section 5 describes the usability analysis of the proposed system.

II. State of the Art

The challenge of creating a virtual prototype of a LCA system pertains to the virtual assembly research area. The relevance of Virtual Reality as a technology for addressing design for assembly tasks was already assessed in the nineties [7]. The Virtual Assembly Design Environment (VADE) [8] was one of the first Virtual Reality Environment for assembly planning and evaluation tasks. The VADE system was the result of a research project started in 1995 with the purpose of exploring the potential and the technical challenges in using VR technologies for design and manufacturing. At the same time, Dewar et al. [9] explored the usage of virtual reality for assembly tasks, focusing of the development of novel tools to simplify the interaction in the virtual environment: more specifically, the proximity snapping and collision snapping tools were proposed by the authors to assist the assembly planner in joining two objects together in a virtual world. A review of VR for assembly methods prototyping shows that after a decade of research the main challenges for virtual assembly tasks were still collision detection, inter-part constraint detection and management of physics-based modeling [10]. Liu et al. [11] focused on physics-based virtual assembly methods as a mean to overcome the limitations of constraint-based virtual assembly methods. Noghabaei et al. [12] developed a platform for training workers through virtual manipulation of building objects in an IVE. A worker wearing a VR headset can perform an assembly of multiple virtual modules, thus identifying potential problems and discrepancies of as-built elements. Sharpa et al. [13] researched a collaborative virtual assembly environment to demonstrate both assembly and 3D presentation simultaneously for an aircraft. Pan et al. [14] focused on the training capabilities of virtual simulations, developing a virtual assembly system of robotic parts in an IVE, to train students in assembling robotic parts using interactive, virtual equipment as in the real environment. Overall, virtual assembly applications have been proven to represent a powerful tool to evaluate and plan product assembly, as well as for training activities and collaborative prototyping.

III. Contribution

This paper investigates the usage of an IVE as a tool for rapid and easy prototyping of LCA solutions. The proposed system loads from a digital library the 3D models of the components available to build a LCA system, as well as detailed information such as measures, materials, connection points, and physical properties (such as friction). Then, the user can add components in a virtual room created in Unity 3D[1] and easily join them together through a magnetic snap mechanism. The virtual room can contain a DT of the real production line or work area, depending on the user's needs, and/or a LCA template to be used as a reference. The system provides a set of tools to speed up the LCA system creation, such as dynamically scaling a component, copying position or size properties among components, and easily combining or separating them. The system has been designed to be used by a VR headset such as the HTC VIVE Pro: the VIVE controllers act as the main interaction tool inside the IVR. When the user completes the creation of the LCA system, it is possible to simulate its physical properties using the Unity Physical Engine. Moreover, it is possible to obtain a list of all the pieces needed to build the prototype and their dimensions, to easily reproduce them in the real world.

IV. The Proposed System

The proposed system aims to simplify and speed up the design phase of LCA solutions through the creation of an IVE for LCA prototyping. The IVE should provide the users the following functionalities:

- exploring a virtual environment through immersive, virtual reality headsets; the virtual environment may consist of a DT of an existing production line;
- interacting with the elements in the environment; depending on the objects in the virtual scene, realistic behaviour such as collision, occlusion, or actions such as pick and place should be available to the user;
- adding objects to the scene from an existing virtual warehouse, modifying them and combining them together to create LCA systems. These system requirements should enable users to virtually create LCA solutions.

Figure 1 shows the architecture of the proposed system. The IVE application has been designed with Unity 3D and the SteamVR plugin[2] to deploy a 3D application compatible with the most recent head-mounted displays, such as the HTC VIVE or the Oculus Rift. Since the HTC VIVE Pro[3] is one of the most performing device on the market in terms of technical specifications, it has been chosen for testing the proposed application both during the developing stage and for the usability evaluation. The HTC VIVE Pro, released in 2018, is equipped with two OCLED displays with a maximum resolution of 1600x1400 at 90Hz and a guaranteed field of view of 110°. Moreover, it is possible to deploy the HTC VIVE Pro either with a cabled connection to the computer running the application, thus providing a lighter headset, or with a wireless connection, thus providing fewer constrictions when moving in the virtual and physical space. The HTC VIVE Pro relies on

[1]https://unity.com

[2]https://valvesoftware.github.io/steamvr_unity_plugin/

[3]https://www.vive.com/us/product/vive-pro-full-kit/

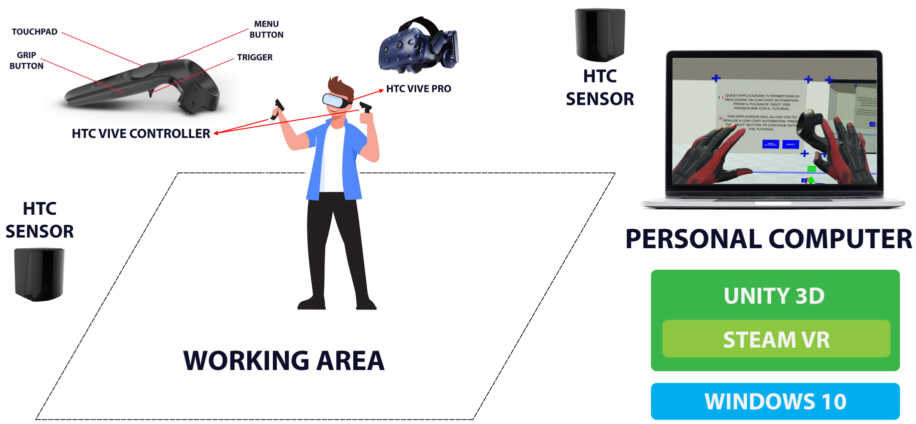

Fig. 1. A schema showing an overview of the proposed system, the software layers, and the hardware setup.

two HTC Base Stations (or HTC Sensors), which consist of an infrared wireless camera that tracks both the headset and the controllers using the Lighthouse technology; these cameras are capable of tracking a room-scale area of up to 4.5 x 4.5 meters, but additional HTC Sensors can be employed to extend the working space. Additionally, the HTC VIVE Pro is equipped with a frontal RGB camera that could be used to provide the user visual feedback of the surrounding environment. To provide easy interaction with the virtual objects the HTC VIVE Pro system includes two HTC VIVE controllers (shown in Figure 3): these controllers are used both to track the user's hands position in the virtual scene and to provide a physical interface by buttons, triggers and touchpad surfaces. SteamVR provides an easy bridge to map all the buttons and actions available through the HTC VIVE controller to the desired actions defined in the virtual environment application. Moreover, Steam VR provides a virtual counterpart of the controllers, thus enhancing the sense of presence provided by the virtual application. Visual feedback is provided on the virtual controller to mimic the interaction performed by the user on the real ones. The functionalities requested by the proposed use case involve mainly two types of virtual objects: the user's hands and the objects available to create the LCA. A third relevant element consists of virtual menus, which extend the available functionalities with respect to the maximum number of inputs available through the controllers and provide also visual information about the system status and possible actions. The main interaction paradigm adopted in the proposed system rely on the user's hands positions in the virtual space. Depending on the system status and the hands positions, different actions are available. The actions mapped on the controller buttons are the following:

- menu button $\rightarrow$ open a contextual menu;
- trigger button $\rightarrow$ grab one object;
- grip button $\rightarrow$ contextual object interaction;
- touch pad $\rightarrow$ contextual object interaction.

The status of the system, which defines the actions available to the user, relies on the concept of collision: the collision system provided by the Unity physic engine identifies starting, ending or ongoing collisions between objects in the scene. To intercept collisions events between the user's hands and the 3D objects in the scene, it is sufficient to attach the collider component on them. The component system is a functionality available in Unity that let the user assign standard behaviours to objects without the need to explicitly write code. Components rely on library-based scripts that could be tuned through a set of variables accessible from the inspector panel of the specific component. In the case of the collider component, depending on which object the user is currently touching and its status, different actions have been mapped to the different inputs provided by the controller. Thus, some actions are contextual since they rely on the position of the hands in the space and if they collide or not with an object; some specific actions requires both hands to collide with the same object.

The menu button available on the controllers allows the user to open a contextual menu: the main menu if no object is touched or the object menu with the available actions otherwise. Once the menu is displayed in the scene, the user can select the different options by pushing the virtual buttons as if they were physical ones.

It is possible to push the trigger to grab the object and change its pose in the scene when the user's hands collide with an object. If both hands grab the same object, depending on the object properties, it may be possible to change its scale simply parting the hands or bringing them together.

One of the main challenges furnishing and assembling 3D tools is to explicitly save the relations between objects apart from their coordinate proximity. For example, if a set of tools is accurately positioned over a table, the system should allow the user to move the table keeping the relative position of the tools respect to the table, because the tools 'belong' to the table. To address this challenge, a dependency algorithm has been developed to handle the relations between objects in the scene.

With these functionalities, the system can enable users to virtually assemble a workstation or production line, starting from a library of 3D models. However, to effectively create a virtual prototype that the user will likely build in the real world, simply putting two parts nearby would not be enough to describe an assembling process, like the one necessary to deploy a LCA, which requires the user to combine different objects together. This challenge can be oversimplified, considering only the objects' absolute and relative positions, and the relations between them. However, the creation of a complex object as a result of combining together simple parts requires some additional rules:

- which objects can be connected to another one;
- which points on the object's surface are valid connection points;
- which objects are compatible with each connection point;
- what is the resulting mesh when two objects are connected.

To address this challenge in the LCA creation process, a library of available parts to construct the LCA has been defined. For each element of the library, it was necessary to create a prefab in Unity containing both the graphical representation of an object and its properties. Unity's prefabs define a standard setup of a virtual object which can be easily added to a scene multiple times. For each prefab, an XML file has been defined to contain all the object's invisible properties, such as:

- the bounding box defining the grabbing area for the virtual hand;
- the list of coordinates of the available connection points;
- the list of compatible connection points from other objects;
- the relative position of one connection point with respect to a compatible one, in terms of relative coordinates (which define the resulting mesh after connecting two objects);
- the list of scaling actions available (if any) if it is possible to change the object dimensions (e.g, it may be possible to scale a pipe along an axis to obtain a longer or shorter one).

Once an object is added to the scene through its prefab, all these data are loaded and applied to the virtual object to provide a specific behavior when the collision events occur. Figure 2 shows the components available to the user to build LCA systems, which comprehend: pipes, different joints, wheels, crates, and the parts needed to build a roller conveyor. One of the available object is a metal pipe, which can be scaled to obtain a longer or shorter pipe, with two connection points at the extremes compatible with the round connection point of the white connector, whereas the 4 sides of the pipeline, along their entire length, are compatible with the narrow connection point of the white connector. When the user grabs one object with each hand and brings them together, the objects are joined (creating a complex object made of more than one basic object) only if all the following conditions are true:

- a collision event between the two objects occurs;
- the two objects share at least one compatible connection point (which has not been already connected to another object).

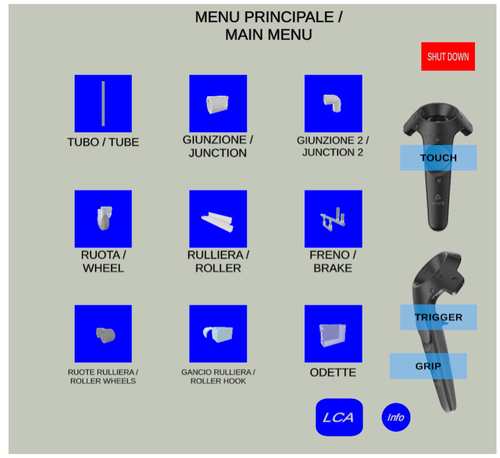

Fig. 2. The menu representing the objects available to assemble the LCA system.

If these conditions are both true and more than one compatible connection point is available, the system connects the two objects automatically selecting the closest available point based on the collision coordinates. Then, depending on the object type (further detailed in each object XML file) and distinguishing between merging two basic objects, adding a basic object to a complex one or merging two complex objects, a father-child relation between the two objects is defined, and the child object is added to the list of children. Finally, exploiting the relative position of one connection point with respect to a compatible one, the resulting mesh after connecting two objects is defined.

The list of children of a given object is crucial to provide the 'separate' functionality to the user: if the user grabs a part of a complex object and want to separate it, the user only needs to press the grip button and move the part away.

Another option available to the user is to change the connection position of one object to another. This is useful if multiple connection angles are available, or if the connection points available for an object define a continuous range (as per the four sides of a metal pipe). In such cases, pressing the touch pad while grabbing an object let the user change the pose along the free axis or angle, locking all the other ones and preventing the part separation. Figure 3 shows the contextual menu that a user can open while touching a simple object. The menu shows the coordinates and dimensions along the three axes for the given object, as well as the available actions:

- changing the scale on one of the three axes;
- deleting the object;
- copying the object dimensions to or from another object of the same type in the scene;
- copying one of the three object's coordinates from another one, to easily align objects along a desired axis.

For a complex object, since the 'copy to' or 'copy from' commands are not feasible, the split command to separate all the parts is available, as well as the distance command, to compute the gap between the current object and another one, which the user can select simply by touching it in the virtual scene.

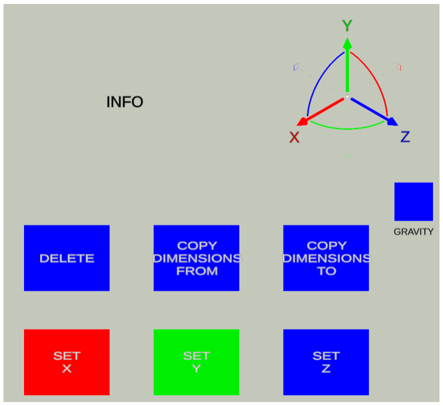

Fig. 3. The contextual menu available while touching a simple object.

When the user completes the LCA system, it is possible to assess its physical behavior: the user can add to the scene crates or other similar objects, then from the main menu is possible to turn on the Unity physic engine. To this end, it is necessary to define the physical properties of each object of the 3D warehouse in its XML file, e.g. weight and friction. Moreover, when the prefab of an articulated object (such as a lever) is created, it is necessary to code its behavior in terms of possible rotations or translations either in terms of physical behavior (for collisions with other objects) or in terms of reactions to user's actions if it is not possible to simulate the physical interaction between the user and the object.

Finally, the system implements save and exit functionalities that allow the user to recover a previous LCA assembly session, simply saving for each object in the scene the pose coordinates, the children list, and the scale factors in an XML file. Moreover, when a user completes the LCA design, the saving function provides a complete list of all the components (and their measures) needed to build the LCA in the real world. After completing the development of the proposed system, a thorough validation step has been carried over to verify that each functionality provided by the system respected the desired behavior.

V. Tests and Results

To assess the usability of the proposed system, a LCA building task has been defined whereas users have to build a LCA solution using a template model for reference, as shown in Fig. 4. The user is first introduced to the system capabilities and the user interface through a step-by-step tutorial in the IVE. Then, the user can freely interact with the system to replicate the proposed LCA solution. The system has been tested by two groups of users for a total of eleven testers with age ranging between 20 and 50 years. Five users belonged to the experienced users' group, representing people that have everyday experience with VR technologies, whereas six users belonged to the inexperienced users' group, representing people with little to no experience with VR technologies. A

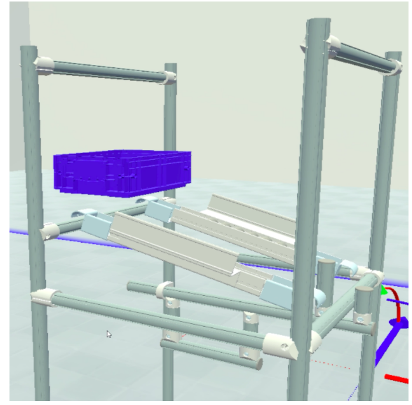

Fig. 4. An example of a LCA system assembled in the proposed IVE through the usage of a VIVE HTC PRO Headset and VIVE Controllers.

video showing one of the users performing the prototyping task can be found at (https://sanna.polito.it/LCA/test.mp4).

The system usability has been assessed by the System Usability Scale (SUS) questionnaire [16], whereas the NASA task load index (NASA-TLX) questionnaire [17] has been used to assess the perceived workload. Completion times have been recorded for further analysis.

TABLE I
SUS Scores, training times and prototyping times for expert users (1-5) and inexpert ones (6-11).

User	SUS Score	Training Time	Prototyping Time
1	95	10:59	16:36
2	80	6:40	10:08
3	92.5	5:34	11:53
4	72.5	11:21	16:00
5	87.5	11:40	21:10
Average	84	09:15	15:09
6	75	10:47	14:45
7	92.5	5:26	18:45
8	67.5	12:38	15:57
9	97.5	04:15	19:02
10	70	09:54	07:25
11	80	08:23	12:46
Average	80.42	08:34	16:27
Total Average	**82.05**	**08:53**	**15:52**

Table I shows, for each user, the SUS Score, the time spent in training with the system and the time spent performing the prototyping task, as well as the total average values. The results of the two groups are quite similar for all the three parameters, showing that the system's usability has been highly appreciated by both groups: the average SUS score for experienced users is 84, with a standard deviation of 9.45, whereas the average SUS score for inexperienced users is 80.42, with a standard deviation of 12.19. It is possible to refer to the rating scale provided by Bangor et al. [15] to understand the obtained results: figure 5 provides a graphical comparison of the adjective ratings and school grading scales, in relation to the average SUS score. Based on the obtained average results, the system has obtained a B grade from both groups, with the experienced group evaluating the system as excellent. Moreover, since both the average SUS scores, training times and prototyping times of the two groups are similar, it is possible to affirm that the system is easily approachable to people experiencing IVR systems for the first time. Finally, all the users of the experienced group were proficient with IVR technologies, whereas the users from the inexperienced group had different levels of proficiency with IVR; thus, even if the average age of the participants of the experienced group was higher than the other one, the resulting SUS scores are similar. To identify any pattern related to the age of the participants it would be necessary to expand the evaluation to at least four groups: older user, experienced in LCA, with (G1) or without (G2) experience in IVR, and younger user without experience in LCA, with (G3) or without (G4) experience in IVR.

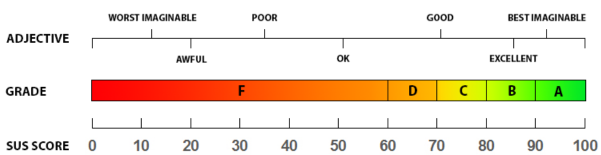

Fig. 5. A graphical comparison of the adjective ratings and school grading scales, in relation to the average SUS score, based on [15].

Table II shows the NASA-TLX diagnostic subscores on a 0-100 scale. The scores show that the most demanding factors of the task's workload are mental demand and effort. This result was expected, since the task was performed in an IVE, which is usually mental demanding despite the task at hand, and requires more effort, since even user experienced with IVR are usually more confident with traditional interfaces. It may be possible to compare the mental workload of the proposed system in both IVR and VR to better understand how the interface impacts on the task, since it does not exist a standard software for LCA design that could be used for comparison. Overall, most of the users were satisfied with their performances in accomplishing the proposed task: this result is confirmed by the NASA-TLX scores, since the frustration value is quite low, and the performance value is also low, which correspond to a good level of satisfaction perceived by the user.

TABLE II
NASA-TLX DIAGNOSTIC SUBSCORES ON A 0-100 SCALE.

Mental Demand	60,91	Performance	22,22
Physical Demand	28,18	Effort	52,73
Temporal Demand	25,00	Frustration	28,18

VI. CONCLUSIONS

The advent of Industry 4.0 has boosted the usage of innovative technologies to promote the digital transformation of manufacturing realities. IVR can provide a whole new level of interactivity and easiness of use for designing and prototyping tasks and can enhance remote collaborations among users. This paper investigates the usage of an IVE as a tool for rapid and easy prototyping of LCA solutions. To assess the usability of the proposed system, a LCA building task has been defined whereas users have to build a LCA solution using a template model for reference. The proposed system has been tested by both experienced and inexperienced users, and preliminary results show that the system's usability has been highly appreciated by all of them, with a SUS average score of 80.42 for the inexperienced group and 84 for the experienced one. Future works will focus on adding new objects to the LCA library, thus expanding the possible tasks to perform in the IVE to further assess the proposed system. Another interesting research goal would be to deploy other types of controllers (such as haptic ones) for interacting in the IVE, and comparing them with the current interface.

REFERENCES

[1] Erboz, G. (2017). How to define industry 4.0: main pillars of industry 4.0. Managerial trends in the development of enterprises in globalization era, 761-767.
[2] Tao, F., Zhang, H., Liu, A., & Nee, A. Y. (2018). Digital twin in industry: State-of-the-art. IEEE Transactions on Industrial Informatics, 15(4), 2405-2415.
[3] Berni, A., & Borgianni, Y. (2020). Applications of virtual reality in engineering and product design: Why, what, how, when and where. Electronics, 9(7), 1064.
[4] Gaudette, D. A. (1998). Low Cost Automation. SAE transactions, 1108-1113.
[5] Katayama, H., Sawa, K., Hwang, R., Ishiwatari, N., & Hayashi, N. (2014, July). Analysis and classification of Karakuri technologies for reinforcement of their visibility, improvement and transferability: An attempt for enhancing lean management. In Proceedings of PICMET'14 Conference: Portland International Center for Management of Engineering and Technology; Infrastructure and Service Integration (pp. 1895-1906). IEEE.
[6] Anggrahini, D., Prasetyawan, Y., & Diartiwi, S. I. (2020, July). Increasing production efficiency using Karakuri principle (a case study in small and medium enterprise). In IOP Conference Series: Materials Science and Engineering (Vol. 852, No. 1, p. 012117). IOP Publishing.
[7] Jayaram, S., Connacher, H. I., & Lyons, K. W. (1997). Virtual assembly using virtual reality techniques. Computer-aided design, 29(8), 575-584.
[8] Jayaram, S., Jayaram, U., Wang, Y., Tirumali, H., Lyons, K., & Hart, P. (1999). VADE: a virtual assembly design environment. IEEE computer graphics and applications, 19(6), 44-50.
[9] Dewar, R. G., Carpenter, I. D., Ritchie, J. M., & Simmons, J. E. (1997, July). Assembly planning in a virtual environment. In Innovation in Technology Management. The Key to Global Leadership. PICMET'97 (pp. 664-667). IEEE.
[10] Seth, A., Vance, J. M., & Oliver, J. H. (2011). Virtual reality for assembly methods prototyping: a review. Virtual reality, 15(1), 5-20.
[11] Liu, K., Yin, X., Fan, X., & He, Q. (2015). Virtual assembly with physical information: a review. Assembly Automation.
[12] Noghabaei, M., Asadi, K., & Han, K. (2019). Virtual manipulation in an immersive virtual environment: Simulation of virtual assembly. In Computing in Civil Engineering 2019: Visualization, Information Modeling, and Simulation (pp. 95-102). Reston, VA: American Society of Civil Engineers.
[13] Sharma, S., Bodempudi, S. T., Arrolla, M., & Upadhyay, A. (2019, December). Collaborative Virtual Assembly Environment for Product Design. In 2019 International Conference on Computational Science and Computational Intelligence (CSCI) (pp. 606-611). IEEE.
[14] Pan, X., Cui, X., Huo, H., Jiang, Y., Zhao, H., & Li, D. (2019, November). Virtual Assembly of Educational Robot Parts Based on VR Technology. In 2019 IEEE 11th International Conference on Engineering Education (ICEED) (pp. 1-5). IEEE.
[15] Bangor, A., Kortum, P. and Miller, J. (2009). Determining what individual SUS scores mean: Adding an adjective rating scale. Journal of usability studies, 4(3), pp. 114-123.
[16] Brooke, J. (1986). System Usability Scale (SUS): a "quick and dirty" usability scale. Usability Evaluation in Industry. Taylor and Francis.
[17] Hart, S. G. (1986). NASA task load index (TLX).