



Knowledge Management in Open Industrial Virtual Reality Applications

Filip Górski^(✉), Paweł Buń, Przemysław Zawadzki,
and Radosław Wichniarek

Chair of Management and Production Engineering,
Poznan University of Technology, Poznań, Poland
filip.gorski@put.poznan.pl

Abstract. This paper presents practical results of a new, open approach to knowledge formalization and management in Virtual Reality applications for use in industry, especially in design and training purposes. Idea of applications with open access to knowledge is presented. Three cases of VR systems are shown – two for the design and one for training, all implemented in industrial companies in Poland. The focus is put on concepts of storing and easy editing knowledge in the applications, in a way to ensure self-sufficiency of applications in a given company after deployment. Dedicated tools for knowledge edition are presented, along with selected results of their implementation in scope of effectiveness of expanding the knowledge base.

Keywords: Virtual reality · Design · Training · Manufacturing

1 Introduction

Modern production companies constantly try to find ways to be more competitive. Today, clients want to have a share in decision-making about shape, functions and appearance of their products and want them delivered as quickly as possible. That is why increasing competitiveness of manufacturing companies nowadays consists mostly in offering individualized, customized products, delivered in a very short time. This is known as the mass customization (MC) approach [1].

The new trend of Industry 4.0 focuses on smart factories, Internet of Things and cloud computing, as well as human-machine integration on every possible level. This assumes digitization of virtually all aspects of production – a concept known previously as digital manufacturing or computer integrated manufacturing. It requires to translate the design and production processes into a virtual space, what is possible thanks to use of the virtual technologies (recently described as XR) – virtual reality (VR), augmented reality (AR) and mixed reality (MR).

Virtual reality has been utilized industrially for few decades now. First implementations of VR in mechanical engineering come from the 90s of XXth century, e.g. in Boeing company [2]. Since the beginning, two aspects were highly emphasized – use of VR for aiding the design of new products and virtually assisted training of various manufacturing and auxiliary processes. There are many successful industrial implementations of VR in both of these aspects, in companies such as Volkswagen [3, 4],

MAN [5] or Ford [6]. Efficiency of VR relies mostly on the immersion effect, what is a human feeling of being present in an artificially created world. The VR simulations are human-centric, focusing on stimulating the senses, particularly sight, hearing and touch, in order to persuade one's brain into believing of being present elsewhere. This aids imagination, allowing more efficient training or design [7, 8]. Recent revolution in both software and hardware, such as introduction of cheap VR headsets, e.g. Oculus Rift, drastically decreased prices of VR systems, allowing its accessibility for even the small and medium enterprises. Many new companies offer development of VR software applications, due to increasing market demand. The accompanying technologies of augmented reality (AR) and mixed reality (MR) also gained considerable impact and a number of applications.

Requirements of production companies, especially considering the mass customization and Industry 4.0 premises, make traditional ways of development of VR (or AR/MR) software obsolete. Most VR solutions are built without any methodology whatsoever [9] and the most important hindrance, industry-wise, is closed nature of such applications. It means that knowledge about a given product or process is hard-coded into the application and any changes (for example, in product shape or sequence of a process) is not possible from the viewpoint of a manufacturing company, as it requires going back to the source code, meaning costly and lengthy intervention back from the original developer.

This paper presents one of the aspects of a methodology of building open VR systems and applications – knowledge formalization and management in industrial VR apps. Idea of open-access VR application is presented, along with two cases of systems with a library of formalized knowledge, assisted with special, dedicated tools for knowledge editing from inside the original user – a production company.

2 Methodology

2.1 Idea of Open Industrial VR Applications

Two main purposes of use of VR techniques in a modern production company can be distinguished as below:

- virtual design and prototyping – creation of virtual models of products and processes in a company, especially when introducing new ones, allowing easy configuration and checking of features of particular design in an immersive visual environment [7, 10],
- virtual training – using the aforementioned virtual models for quick and safe learning of skills and procedural knowledge by employees of a given company [11, 12].

In general, there is a requirement of VR applications to be maintained throughout the whole lifecycle of a given product or a process. The maintenance is ideally done inside the company, without need of hiring external developers. Access to knowledge about process or product should be open and data should be easily changed without programming. Currently used approaches in development of VR applications are derived from development methodology of video games, thus not allowing flexibility

and openness on a level required by the industry. Additionally, there are no methodologies of building VR (or AR/MR) applications for industry, as mentioned by many authors [9, 13, 14]. Existing methodologies regarding VR in manufacturing are scarce or limited [15, 16]. No methodology allows building open VR applications – usually it means that after coding, any access to product/process knowledge is lost to everyone but the programmer.

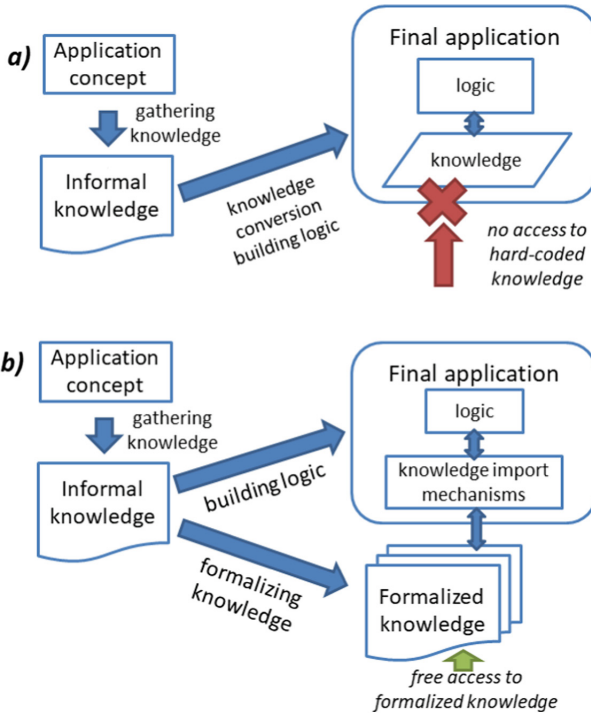


Fig. 1. Traditional (a) and new (b) process of building VR applications for engineering [17]

An innovative approach to this problem is proposed, on the basis of a range of realized projects and implementations of VR solutions in industrial companies – a knowledge-based approach. The most important change from the traditional approach is focusing on recording the product/process knowledge in an application in a formal way and storing it outside the application, for later easy access both by humans and in-built algorithms. The key idea is presented in Fig. 1 and was described in earlier publications [17].

The proposed methodology of building open VR applications – Knowledge Oriented Methodology of Development of Virtual Reality Systems (KOMDEVRS) is based on Knowledge Engineering and Design Thinking approaches. The main stages of development are presented in Fig. 2.

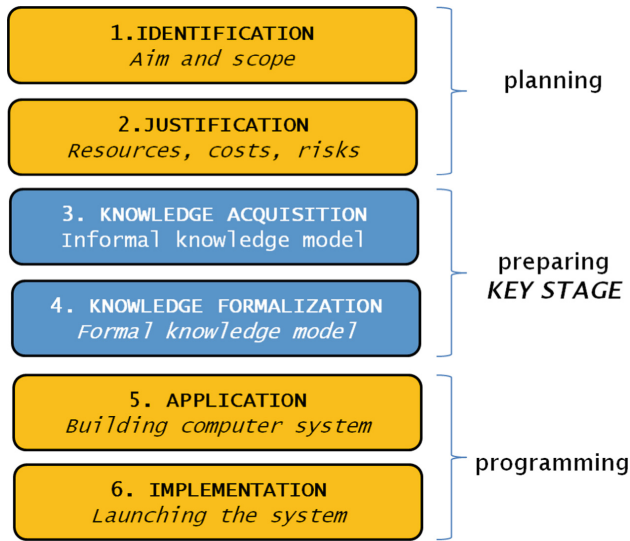


Fig. 2. Basic stages of the proposed methodology (KOMDEVRS) [17]

The knowledge acquisition and formalization stages are the key stages, at which a VR system gains its unique capability of openness (flexibility). Knowledge acquisition is aimed at obtaining all the necessary knowledge in an unstructured (unordered) way, in all forms possible (2D, 3D, audio, text, etc.). Knowledge formalization consists in structuring the knowledge, organizing it and assuming a certain way of recording various units of the knowledge, from the viewpoint of a specific application. The further part of the paper briefly overviews basics of knowledge formalization and management in open VR applications for industry.

2.2 Knowledge Formalization

In the case of building open VR systems according to the proposed methodology (Fig. 2), knowledge formalization in a technical sense is a process leading to creation or development of a basic module of the whole VR system, which is the library (also known as the knowledge base). It can be also related to creation of a second module, which is the application for knowledge management. The main module – a visualization module – could be created later, as a third, final module, when all the knowledge management mechanisms are settled down. However, usually a simple non-open proof of concept of a working VR application has to be created prior to main development (at the justification stage, for example to convince the stakeholders), hence in the chronological sense, the client side application comes to life first. What is noteworthy here, is that in case of the closed VR applications created in a traditional manner, there is only one module – the client side module and the knowledge base is hidden inside its code.

Focusing on the knowledge formalization process, it should be divided into the following phases:

1. Creating a base structure of the open knowledge base, meaning the folder tree, with regards to specific forms of knowledge (known from the previous stage – knowledge acquisition) and recorded data. The folder structure should be created with future expansions in mind and is also a product of decisions on the formalization level and type (described further).
2. Making decisions on file formats stored in the knowledge base, depending on the form of the knowledge and available data import/export mechanisms in a selected VR development software (examples: 2D data – JPG, PNG, 3D data – OBJ, STL, FBX, STP, metadata – TXT, XML etc.).
3. Creating templates of names of data files, with regards to future scalability of the knowledge base. Making a decision on free or constrained naming of appropriate files.
4. Creating a list of file types saved in the knowledge base, along with determination of their format, function, naming and location.
5. Development of structure of metadata files (plain text or special format, such as XML) – determination of methods of references to particular data files, definition of language of logic describing instances of objects in the VR application.
6. Filling the folders with proper data files, in quantity allowing to prepare the full application, along with validation of procedure of knowledge base editing and expansion.
7. Manual filling metadata files, converting informal knowledge about the product and/or the process into description of data allowing the VR application to run its logic realistically.
8. Creation of algorithms of import, export and processing of knowledge stored in the base by the visualization module of the future VR application.

Such a base of knowledge, completed with algorithms of import, export and processing of knowledge by the VR visualization module can be a basis for building the proper client side application in a selected VR development software (e.g. Unity or Unreal engine).

The formalization is not always required to be complete, meaning that not all the knowledge is stored outside the main VR application. Sometimes part of the knowledge (particularly 3D objects) can be hard-coded without losing flexibility or at the expense of time needed for implementing the full formalization scheme. This is particularly true for 3D models, which are generally difficult to import and process during runtime of any VR application. That is why a concept of formalization levels must be introduced.

Basic four levels of formalization proposed by the authors pertain to particular forms of knowledge describing given product or process in the VR application. These are the following:

1. Text form – descriptive knowledge (text messages, descriptions etc.)
2. Multimedia form – raster graphics (textures, schemes etc.) and audio/film

3. Text form – knowledge of (a) structure (list of parts – BOM, object positions in 3D space, colors) and (b) operation (options, values and their constraints, process sequence with progress conditions, animations etc.)
4. 3D model form – geometry of product, tooling, environment and characters, along with animations.

Usually level 1 and 2 are easy to achieve by simple means of reading appropriate files from external folders rather than internal resources hard-coded in the application. Level 3 is more difficult, as it requires introduction of dynamic hierarchy, positioning and flexible programmed logic. Level 4 is the most difficult to perform properly, as 3D models in a traditional work scheme are usually known and prepared well before programming other functionalities and certain operations (e.g. texture mapping, lightmap baking) can have limitations if performing them during the runtime.

The knowledge acquired for a given application can be described quantitatively and qualitatively. In terms of quantity, a typical unit of knowledge is composed of a text, a 2D graphics, a 3D model (optionally with animation) and optionally an audio. This can constitute, e.g., a single part or subassembly of a designed machine or a single block in a training program. Qualitatively, description of achieved levels should be marked, as e.g. 1, 2 and 4 meaning that the acquired knowledge contains simple descriptions (text), graphics and 3D models but does not contain metadata on structure and/or operation.

2.3 Knowledge Management Tools

Formalized knowledge management by manual file edition is a labor consuming task, which also required advanced knowledge of application mechanisms and it can also generate malfunctions (e.g. by a simple typo in a name of object inside a metadata file). That is why for effective knowledge management there is a need of building an additional application (module of a system). Its main functionality can be compared to CMS (Content Management Service) applications used for edition of WWW pages. Aims of its work are as following:

- acceleration of process of knowledge edition when reconfiguring or adding new contents to a VR system – by automation of edition of files stored in the knowledge base,
- elimination or serious limiting of risk of making an error while editing formalized knowledge in the application,
- facilitation of management of data exported by the visualization module as a result of performed activities by a user (e.g. product configuration report, virtual training statistics),
- in case of dispersed (network) system architecture, i.e. many remote instances of the visualization module – propagation of the updated knowledge base to all the VR system instances, with use of cloud mechanisms.

Application for knowledge management in an open VR system may have one of the two following forms:

- windowed 2D application without the 3D mode,
- 3D application with a graphical user interface.

The former is good for simple metadata and data management, when there is no need to preview 3D data from inside the management application. The latter is required when advanced 3D scenes are composed in the VR system and lack of 3D preview would not allow their effective placement. The latter type of knowledge management software is well known in the video game industry as the so-called “level editors”.

3 Selected Use Cases and Results

3.1 Presenter of Upholstered Furniture

The Presenter of Upholstered Furniture (PMT) is an application for virtual design and ordering of single pieces of upholstered furniture. The application is of “projective VR” class, with a given product presented on a large screen with touch-based controls. It is meant for use by a client in a retail shop. The application has two main client-side modules: module for selection of product and module for presentation of the selected product (Fig. 3). These modules were created in the EON Studio software.

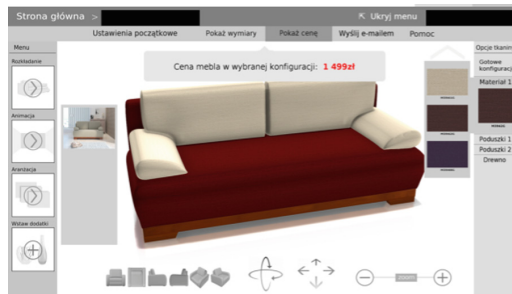


Fig. 3. The presenter of upholstered furniture (PMT) – presentation view

Regards knowledge formalization in the PMT application, it was assumed at levels 1, 2 and 3b, with omitting the levels 3a and 4. This was due to the manufacturer not having 3D models of their products, so all the model preparation was outsourced to the authors of the application, thus allowing hard-coding.

The following knowledge about the product was described in a formalized way:

- names and types of furniture,
- icons of products,
- photographs and movies,
- prices,
- textures,
- colors,
- options and values (amount of options and amount of textures/colors for a given option for a given piece of furniture, with upper limit),
- contents of product specification for ordering (names of colors, contents of message sent to user etc.).

The PMT application folder structure with description is presented in Fig. 4. Figure 5 presents a knowledge description model for automated generation of list of pieces of furniture in the first module of the application. On the basis of code name of a furniture, being direct name of its icon, the algorithm loads appropriate 2D and text data, as well as creates appropriate link after clicking on the furniture, to launch the second module (visualization of a single piece). The other metadata allow sorting and searching for a given furniture or group of it.

Figure 6 presents examples of metadata describing changeability of textures and colors for a given surface of a given piece of furniture. The file contains a list of references to texture files or particular colors and their icons. On the basis of presence of appropriate number of such defined files and their contents, the visualization module generates an appropriate number of dropdown lists in GUI, allowing configuration of a product.

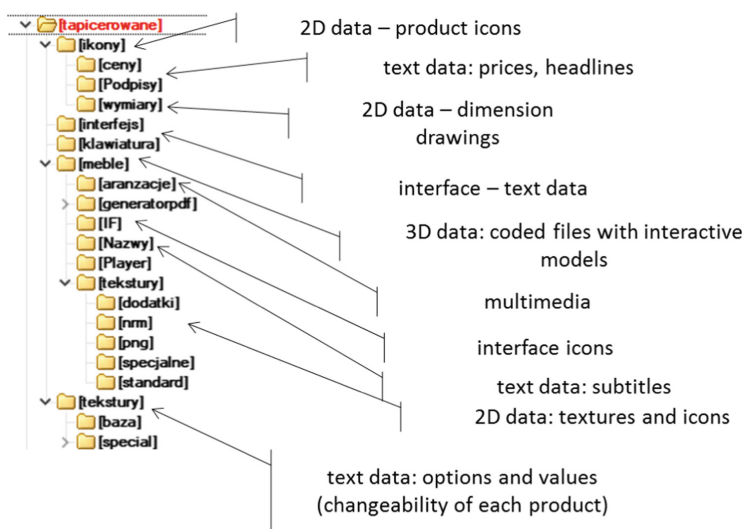


Fig. 4. PMT application folder structure

```

Ida.bmp,w,so,op,now
Ilka.bmp,w,ze,op,now
JanLux.bmp,w,s3,op,now
JuliaLux.bmp,w,na,op,now
JunoLux3DL.bmp,s,s3,op,now
    
```

Fig. 5. PMT application – formalized list of objects (part of metadata file)

Similarly, other interactive elements are defined. A number of metadata text files exist for a given furniture piece, on that basis all the knowledge is presented (names, dimensions, prices etc.).


```

1;1;1,DODATKI\2000_BIALE.JPG,K
0.871;0.788;0.012,DODATKI\0113_ZOLTE.JPG,K
0.780;0.341;0.110,DODATKI\1288_POMARANCZ.JPG,K
0.710;0.086;0.086,DODATKI\0504_CZERNONY.JPG,K
M3991G.jpg,PNG\M3991G.PNG
M3969G.jpg,PNG\M3969G.PNG
M3961G.jpg,PNG\M3961G.PNG
M3960G.jpg,PNG\M3960G.PNG

```

Fig. 6. PMT application – formal description of color changeability (left) and textures (right) for different surfaces of a single object (part of a metadata file)

Structure of the metadata files is strictly formalized, with specific separators used (commas, semicolons, new line markers etc.) along with appropriately shortened folder references. User of the system can decide what products are visible in the application, how are they classified, what can be changed in them, what is their price and what other important information (stored in form of graphics and movies) is attached to them. New textures and colors, as well as multimedia, can be added to the system anytime. Because the formalization is not complete (no level 4), adding entirely new products is only possible from the development level in the software, what was intended. However, the procedure of creating a new virtual product on the basis of its 3D model is simple and it does not require any coding whatsoever, but rather a set of simple operations, such as model import and creating references in appropriate places, as well as changing some metadata inside a visualization file.

Additional 2D application was created using the Visual Basic language, for easy management of knowledge inside the application and its propagation into many instances of the system present in retail stores. The so-called administrative application (PMT Administrator) is presented in Fig. 7. It allows changing virtually all the

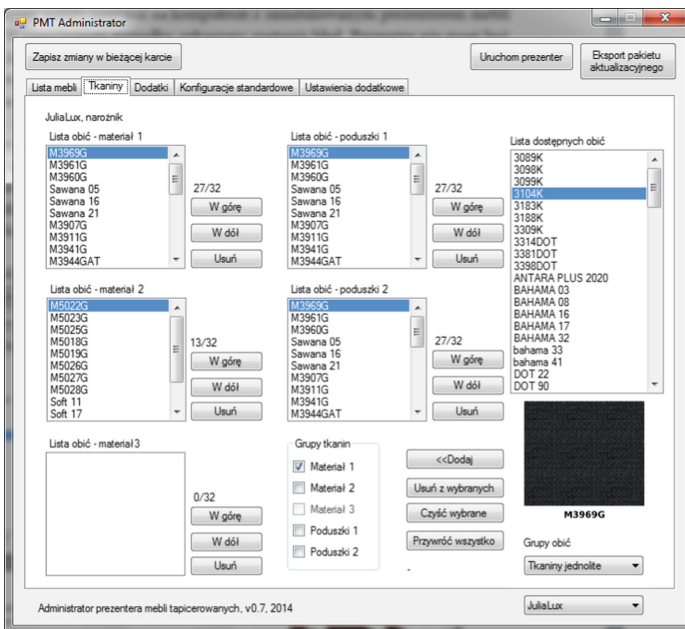


Fig. 7. Knowledge management tool in the PMT application

formalized knowledge, but in a user-friendly manner, not allowing to introduce any mistakes that would generate an error.

Studies on the PMT application allow to state, that it allows generating few hundred to few billion of different variants of a given product (a single variant means a unique specification). Single variant takes less than a minute to prepare. Therefore, a new quality was obtained, which is capability of rapid generation of visualization and specification for ordering, what makes a process of building new individualized variant of a product easier on the manufacturer's part – there is no need of manual creating of variant documentation by a salesman. The application was implemented in retail stores throughout Poland, with an initial package of 39 products and more than 200 different textures and colors. The development process took approximately one year, with a team of five people. Over the course of application run in the stores, capability of efficient adding of contents was proven, as the application was later expanded by 29 new products and approximately 40 textures, in a time of three months, realized by two people: one of original developers and one untrained student, both using the administrative application. Further development work also focused on creating and implementation of a prototype system of design of wooden furniture, intended for internal use in the company [18].

3.2 Virtual Skill Teacher

The Virtual Skill Teacher (VST) is a VR platform for learning practical, technical skills, usually related with operating working vehicles (such as the forklifts, which are the main case), machines, realizing tasks in a certain production process, etc. The key ideas are as following [19]:

- the course is the main entity, it is divided into lessons and exercises, which are further divided into blocks,
- lessons are theoretical and contain a number of graphical, text, audio and interactive 3D info presented in immersive space,
- exercises are sets of practical tasks – they require certain actions in a specified time (such as: drive from place A to B or bring object from one place to another and use it properly) in order to be passed,
- lessons and exercises are divided into blocks, containing certain, distinguished forms of knowledge (described below).
- each course has a set of questions, the exam mode is in-built and also performed in the immersive space.

The contents of lessons have the following form:

- text – descriptions of problems, especially regarding information difficult to present in a visual form (names of standards, formulas etc.), as well as instructions,
- audio – complementary to the text content, read by a lector and launched by interaction with a character of virtual teacher, who is always present in the virtual space,
- 2D graphics – pictures, schemes, diagrams, infographics illustrating blocks,

- 3D content – interactive models of objects (machines and devices, elements of work environment etc.), along with animations, illustrating a current block.

The most important constant elements of the lesson scene are:

- virtual teacher – an animated character helping the user, replacing a living teacher,
- flipchart – space in the center of the scene, where 2D content is displayed,
- environment – e.g. a hall – 3D background space, that the user can navigate.

The user can move freely using three means: on own legs (only short distances, due to HMD cable length and workspace limitation), by use of joystick and by use of teleporting points, located in the environment. Interaction is done by gaze & click mechanism or by using a gaming driving wheel to move the vehicle in the practical exercises. Figure 8 presents a screen from the finished application, showing the most important elements of a single lesson.

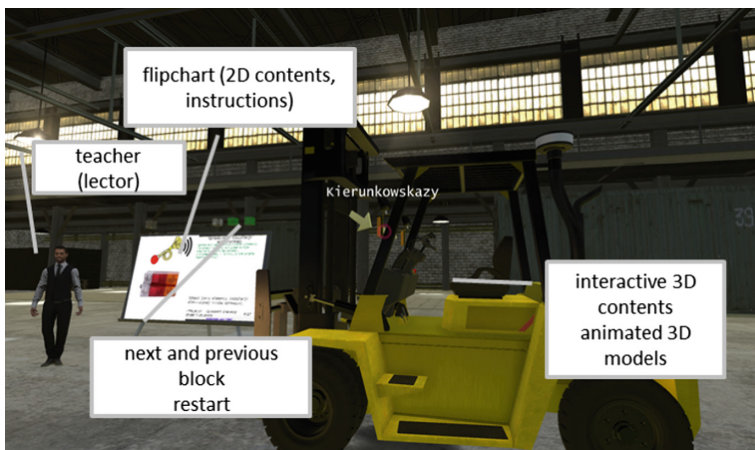


Fig. 8. Basic elements of a lesson in the VST system

The application, made in Unity engine, is thought of as a platform, meaning that it can be complemented with other courses. The current version contains only one course (forklift) but other courses are planned (welding, gantry crane operation etc.). The application studies allowed to state that it is very effective, allowing 90% of shortening of learning time compared to the original, traditionally performed course [19].

The VST system allows flexibility in content management – it allows easy edition and adding new knowledge, i.e. new courses, lessons etc. with minimal or no programming skills necessary. The access to most knowledge is open and metadata are stored in form of text files of open structure. Exemplary structure of an exercise block is presented in Fig. 9. A separate Windows application (VST Administrator) was built for knowledge management. The formalization level is similar to the previous case – levels 1, 2 and 3b were assumed, with 3a and 4 levels omitted. The level 4 was unnecessary in the first version of the application, but it will be introduced in further

installments (the application is still in the development phase – 3 prototypes were created and tested so far).

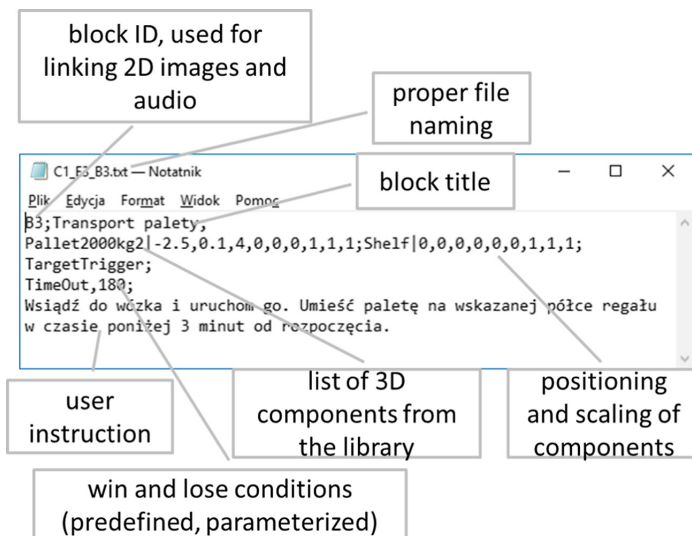


Fig. 9. Structure of formalized knowledge in the VST system [19]

The studies on effectiveness of knowledge management in the application consisted in two phases:

1. Building prototype 1 containing approx. 40% of contents of the application, without special knowledge management tools.
2. Building prototype 2 containing full contents of the application, with use of dedicated knowledge management tools.

In total, the forklift operator course consists of 9 theoretical lessons and 4 practical exercises. This makes for total of 154 knowledge blocks. Prototype 1 contained 3 lessons and 1 exercise, which makes for total 56 knowledge blocks. The team assigned to building both prototypes was the same and consisted of 2 programmers and one 3D graphics specialists, with consultations with two experts (instructors). The results regarding time of building the system by blocks are presented in Table 1, while total time consumption of building the system is presented in Table 2.

Table 1. Knowledge block adding – efficiency ratio comparison in the VST system.

Phase	No. of blocks	Work days	Efficiency ratio (blocks per day)
Prototype 1	56	75	0.75
Prototype 2	98	25	3.92
Sum/mean	154	100	1.54

Table 2. Time consumption of building the VST system.

Stage	Activity	VR programmer work days	3D graphics specialist work days
1	Acquisition of knowledge	5	20
2	Formalization – 1	10	3
3	Implementation (programming & tests) – 1	60	2
4	Formalization – 2	15	3
5	Implementation (programming & tests) – 2	5	2
6	Final improvements	10	-
Total:		105	30

The results comparison between prototype 1 and 2 clearly show, that introduction of knowledge management procedures (namely, formalization of knowledge and dedicated tools for simple knowledge edition) allowed to increase rate at which subsequent blocks of knowledge are added to the application. Advantages of open access to knowledge are clearly visible when comparing times of stages 2 & 3 with stages 4 & 5 (twice more content done in three times shorter total time). Apart from this, entirely new quality was obtained, of open access to courses, lessons and their blocks, for future edition or adding new courses outside the programming engine.

4 Summary

The new approach to storing and editing knowledge in VR applications created for the production industry, presented in the paper, was proven to be effective. It requires changing some of traditional paradigms of VR application development process, but in exchange a capability of building open, manageable and easily extendable solutions is reached. These solutions can be effectively implemented in production companies to routinely support processes of design, production and employee training.

The proposed formalization approach requires considerably more programming in the initial stages of development, so the presented ideas should be put into practice only when there is a direct need for it and scale of a given system (number of products, variants, complexity of processes etc.) justifies it – simple one-time visualizations should be still made in a traditional way, with hard-coded knowledge. The key to proper implementation of industrial VR is always performing of identification and justification stages, that should answer the question of what type of application is needed, who will use it and to what degree can it help.

The authors have already made and implemented a number of open VR systems in various industries. The further research, carried out continuously, focuses on building more complex solutions, with full formalization, and testing their prolonged use in given industries, checking for efficiency and improving various aspects of the presented methodology.

Acknowledgments. The research work reported here was partially financed by Polish Ministry of Development (UE funds: Operational Program Knowledge Education Development) and Polish Ministry of Science and Higher Education.

References

1. Fogliatto, F.S., da Silveira, G.J., Borenstein, D.: The mass customization decade: an updated review of the literature. *Int. J. Prod. Econ.* **138**(1), 14–25 (2012)
2. Tanner, S.: The use of virtual reality at Boeing’s Huntsville laboratories. In: *Proceedings of IEEE Virtual Reality Annual International Symposium* (1993)
3. Hayden, S.: Volkswagen Group to Train 10,000 Employees in VR This Year, Road to VR (2018). <https://www.roadtovr.com/volkswagen-group-to-train-10000-employees-in-vr-in-2018/>
4. Brake, T.: Virtual Reality Collaboration at Volkswagen, Virtual Gurus (2017). <http://www.virtualgurus.com/insights-and-resources/virtual-reality-collaboration-volkswagen>
5. Obermair, F.: MAN and the University of Applied Science Upper Austria Design a Truck Assembly Line using ESI’s IC.IDO - ESI Success Stories, pp. 1–2 (2015)
6. de Sa, A.G., Zachmann, G.: Virtual reality as a tool for verification of assembly and maintenance processes. *Comput. Graph.* **23**(3), 389–403 (2001)
7. Jimeno, A., Puerta, A.: State of the art of the virtual reality applied to design and manufacturing processes. *Int. J. Adv. Manuf. Technol.* **33**(9–10), 866–874 (2007). <https://doi.org/10.1007/s00170-006-0534-2>
8. Martín-Gutiérrez, J., Mora, E., Añorbe-Díaz, B., González-Marrero, A.: Virtual technologies trends in education. *EURASIA J. Math. Sci. Technol. Educ.* **13**(2), 469–486 (2017)
9. Gulec, U., Yilmaz, M., Isler, V.: A literature survey: is it necessary to develop a new software development methodology for virtual reality projects? *J. Univ. Comput. Sci.* **23**(8), 725–754 (2017)
10. Cecil, J., Kanchanapiboon, A.: Virtual engineering approaches in product and process design. *Int. J. Adv. Manuf. Technol.* **31**(9–10), 846–856 (2007)
11. Choi, S.S., Jung, K., Noh, S.D.: Virtual reality applications in manufacturing industries: past research, present findings, and future directions. *Concurr. Eng.-Res. Appl.* **23**(1), 40–63 (2015)
12. Abulrub, A.H., Attridge, A., Williams, M.A.: Virtual reality in engineering education: the future of creative learning. *Int. J. Emerg. Technol. Learn.* **6**(4), 4–11 (2011)
13. Wang, X., Ong, S.K., Nee, A.Y.C.: A comprehensive survey of augmented reality assembly research. *Adv. Manuf.* **4**(1), 1–22 (2016)
14. Palmarini, R., Erkoyuncu, J.A., Roy, R., Torabmostaedi, H.: A systematic review of augmented reality applications in maintenance. *Robot. Comput.-Integr. Manuf.* **49**, 215–228 (2018)
15. Calvo, I., Lopez, F., Zulueta, E., Gonzalez-Nalda, P.: Towards a methodology to build virtual reality manufacturing systems based on free open software technologies. *Int. J. Interact. Des. Manuf. IJIDEM* **11**(3), 569–580 (2017)
16. Gorecky, D., Khamis, M., Mura, K.: Introduction and establishment of virtual training in the factory of the future. *Int. J. Comput. Integr. Manuf.* **30**(1), 182–190 (2017)
17. Górski, F.: Building virtual reality applications for engineering with knowledge-based approach. *Manag. Prod. Eng. Rev.* **8**(4), 64–73 (2017)

18. Zawadzki, P., Górski, F., Buń, P., Hamrol, A., Kuczko, W., Wichniarek, R.: Concept design of furniture with use of virtual reality applications and knowledge based engineering (in Polish). *Innovations in Management and Production Engineering 2016*. Publishing House of the Polish Society for Production Management (2016)
19. Górski, F., Zawadzki, P., Buń, P.: Virtual skill teacher – platform for effective learning of technical skills. In: *Proceedings of the International Conference of the Virtual and Augmented Reality in Education (2018)*