Management and Production Engineering Review
Volume 8 • Number 4 • December 2017 • pp. 64–73
DOI: 10.1515/mper-2017-0037

DE GRUYTER OPEN

mper

BUILDING VIRTUAL REALITY APPLICATIONS FOR ENGINEERING WITH KNOWLEDGE-BASED APPROACH

Filip Górski

Poznan University of Technology, Chair of Management and Production Engineering, Poland

Corresponding author:
Filip Górski
Poznan University of Technology
Chair of Management and Production Engineerings
Piotrowo 3, 60-965 Poznan, Poland
phone: (+48) 61 665 2708
e-mail: filip.gorski@put.poznan.pl

Received: 24 October 2017
Accepted: 10 December 2017

Abstract
The paper presents a novel methodology of building industrial Virtual Reality applications with use of a knowledge-based approach. Virtual Reality is becoming more and more widespread in engineering applications. However, most solutions are immediate and not flexible, especially in maintenance. Traditional way of programming VR applications makes all the knowledge about a product or a process hard-coded, effectively losing access to it from the outside of the programming software. Besides, making new solutions without any methodology whatsoever makes the process longer and less effective. The author proposes to use general rules of available Knowledge Engineering methodologies in order to make the process of building VR applications more effective and to ensure their flexibility and access to stored knowledge, even after an application is deployed. The presented methodology is supported with practical case studies.

Keywords
Virtual Reality, Knowledge Engineering, design.

Introduction

Virtual Reality (VR) systems allow their users to explore and interact with elements of artificially created, three-dimensional worlds in real time [1]. During the last decade, the VR applications improved their level noticeably, from simple applications with non-complex graphics to graphically advanced and logically complex environments, allowing trainings, design aid or decision making in scope of engineering work [2]. It is now possible to obtain a good level of immersion, which is a feeling of being a part of a virtual world [3]. This phenomenon is induced both by use of stereoscopic display systems (e.g. Head-Mounted Display devices) and by use of natural methods of interaction (e.g. gesture recognition). In the recent few years, low-cost (consumer) VR solutions have emerged, which makes availability of hardware such as gesture recognition systems or HMD goggles much higher than several years ago. The software for preparation of Virtual Environments has also undergone rapid development in terms of ease of use, price and capabilities. As a consequence, in author's opinion, building an effectively working VR system for professional use is today easier than ever.

Nowadays VR is mostly used in engineering – for extended, CAD-based virtual design [4, 5], industrial training [6, 7], simulation of machine operation or manufacturing processes [8] and many other applications. One of the industry branches using the VR benefits to a great degree is the automotive branch [9]. The VR has also plenty of applications in medicine and biomedical engineering [10, 11] and general education [12]. VR, as well as Augmented Reality (AR), plays an important role in Factories of the Future, within the Industry 4.0 concept [13, 14].

Despite great advance in development of VR solutions, in author's opinion, there is still a problem of implementing the VR applications in industrial companies. The main hindrance stopping VR from being commonly applied is that most solutions are immediate, created just for the current context purpose,

without use of any methodology that could make the building process faster and obtain more effective results. There is no known dedicated methodologies of planning, building and implementing educational VR solutions in medicine. As each solution is immediate, the time and costs are high, with a frequent number of mistakes and corrections. Another problem is that the VR solutions created in such a way are not suitable for long-time maintenance and they quickly cease being up-to-date. As the industrial VR programming is frequently outsourced, maintaining the application means extra costs, that are not desirable.

The author proposes a novel concept of knowledge-based approach to building VR applications, on the basis of his experience in building VR systems for various industrial companies. This approach, presented in the paper, helps making VR implementations faster and makes them manageable within the company, without need of getting back to the source code. The paper presents basic information on how this approach works, along with some case studies.

Virtual Reality systems in engineering

Industrial VR applications

The basics of a knowledge-based approach to industrial VR applications is to classify them by type of contained and presented knowledge about the product or the process. Such a classification, proposed by the author, is presented in Fig. 1.

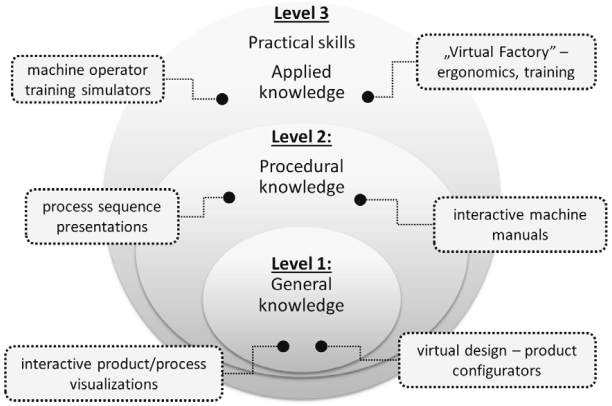

Fig. 1. Levels of knowledge of industrial VR applications

In general, there are two main purposes of use of VR techniques in a modern production company:

- 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 [15],
- virtual training – using the aforementioned virtual models for quick and safe learning of skills and procedural knowledge by employees of a given company [16].

The virtual design and prototyping applications usually belong to level 1, sometimes 2, according to the Fig. 1. The virtual training solutions usually belong to level 2 or 3. This means higher requirements towards interaction techniques, fidelity of visualization and also means more complex logic of a simulation.

The most frequent type of application from level 1 are product configurators. There are numerous of successful implementations of such applications, including configurators of cars [17], public transport vehicles [18], furniture [19] and other products.

In terms of virtual training , there are plenty of various solutions, based on immersive training [20, 21] – these usually belong to level 2. Applications of level 3 often make use of tactile interaction techniques, such as haptic devices [22] and tangible user interfaces (TUI) [23]. Examples of various engineering VR solutions are presented in Fig. 2.

Fig. 2. Examples of industrial VR solutions [17, 20].

Research problem – requirements and current ways of building VR applications

For a Virtual Reality application to continuously function in a specific company, regardless of nature of the problem solved by the application and its tar-

get user group, there is a certain number of requirements that need to be fulfilled, similar to requirements towards other IT systems in a company, used for gathering product or process knowledge, especially in visual terms. They can be generally formulated as following [24]:

- graphics quality,
- easiness of operation,
- use of known peripheral devices,
- data import formats,
- manageability of structure and functions of the application,
- price of implementation and maintenance.

In general, industrial companies require their applications to be maintained through the lifecycle of a product or a process, to which they pertain to. The maintenance is required to be easy and possibly done inside the company, without need of hiring external specialists. Moreover, applications must be flexible – for example, if a new product variant is designed by the design engineers, its features should be incorporated in its corresponding VR configurator as quickly as possible and, if possible, by internal means of the company. If the process changes, its VR training application should be also able to be easily updated with new workplace arrangement, new tools etc. The flexibility also concerns used hardware – if currently used hardware is discontinued, the application should be adaptable to any new hardware that can be introduced in the future (this pertains mostly to Head Mounted Devices and interaction devices, such as motion controllers and joysticks).

The above mentioned capabilities should be, ideally, available to be achieved within a company, without need of external, time-consuming programming processes, but instead by a work of company employees trained in use of a given VR system. Such an approach is not generally practiced in building typical VR applications. There are special industrial VR platforms, such as VRed [25], fulfilling this criterion, except the price (they are very expensive). However, they have a limited range of programming functionalities.

Traditionally built VR applications (using engines such as Unity, EON Studio or Unreal Engine) have a closed form and any changes within the visualization structure are not available without a set of programming tools and an appropriate, detailed knowledge about the programming language and specifics of the solution itself. This contradicts a requirement of flexibility and manageability, as well as external data import. For example, a traditionally made VR product configurator requires going back to the source code when a new set of physical or visual features of a product (such as new colors, textures or parts) must be added. This is a major hindrance and – in author's opinion – one of the main reasons why industry is reluctant in mass implementation of VR solutions.

Current, informally used methodologies of building VR applications for industry do not assume any changes in knowledge contained inside the application from the outside, except going back to the source code. In terms of knowledge about the product or process, it means that immediate access to its base form is lost once the application is deployed. In the further part of the paper, the author proposes a framework methodology and some ideas on how to overcome this problem.

Methodology of KE-aided building of VR applications

Concept of knowledge-based Virtual Reality applications

As mentioned before, the biggest fault of today's VR applications, from the viewpoint of industrial implementation and maintenance, is lack of open access to the knowledge contained within an application after it is deployed, requiring getting back to the code in order to introduce even the slightest changes. This means that, for instance, a simple change such as extending a range of available shell colors in a car configurator, requires a VR programmer to be involved, which generates unnecessary costs and logistic problems. Moreover, the applications are usually made without following any specific methodology whatsoever, making the process far less effective than it could possibly be.

The author proposes an innovative approach to this problem, on the basis of a range of realized projects and implementations of VR solutions in industrial companies – a knowledge-based approach. There is a number of Knowledge Engineering methodologies, helping to build Knowledge-Based Engineering solutions (KBE) [26]. VR applications can be built by following certain guidelines from some of these methodologies. 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. 3 and key differences between the traditional and the new approach proposed by the author are shown in Table 1.

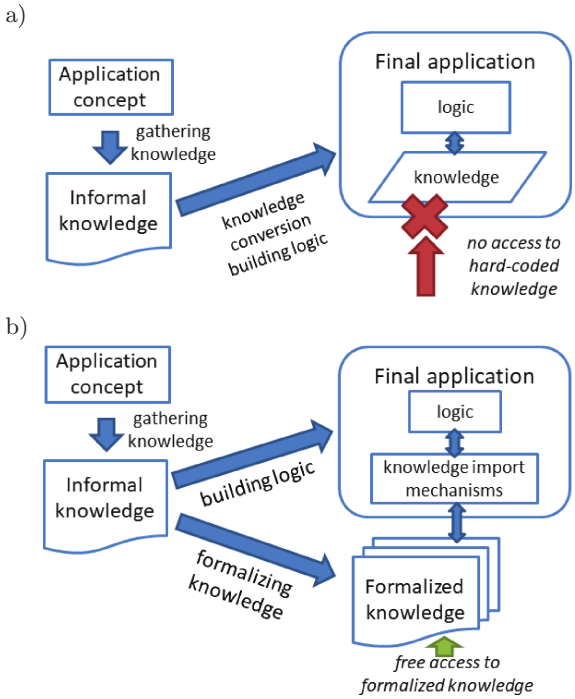

Fig. 3. Traditional (a) and new (b) process of building VR applications for engineering.

Table 1
Key differences between the traditional and knowledge-based approach.

Feature	Approach	
	Traditional	Knowledge-based
Knowledge storing	Hard-coded inside application	Formalized, stored outside application
Knowledge editing	Source code editing	Use of external standard editors for text, image, audio and 3D files
Initial programming effort	Standard	Extended – knowledge import mechanisms required
Expanding knowledge base effort	Labor consuming – back to source code	Short – no programming involved
Knowledge access	Only VR programmers	Anyone

Knowledge Engineering technologies

Knowledge Engineering (KE) is a discipline that deals with problems of acquisition, representation and application of knowledge in computer systems. It is a discipline related to creation of knowledge bases and use of semantic technologies to process knowledge by computer systems [27]. It is comprised of:

- identification of knowledge sources,
- acquisition (obtaining) of knowledge,
- representation of knowledge,
- analysis of identified knowledge,
- creation of bases and repositories of knowledge,
- searching, accessing and sharing knowledge.

Efficient use of knowledge in the design process requires its appropriate acquisition and proper recording in a knowledge base. Methods of knowledge gathering are also dependent on the sources, which can be of variable quantity and form. The most important knowledge sources are:

- personal notes,
- technical documentation (drawings, schemes),
- information from previous projects,
- study results (e.g. from simulation or operation studies),
- libraries of documents, books, papers,
- standards and catalogs,
- remarks from research centers and patent offices,
- remarks from customers and suppliers.

Using the gathered knowledge in a computer system requires building a Knowledge-Based Engineering (KBE) solution, also known as the Knowledge Based System. Over the past two decades, many methodologies have appeared, aiding the development of Knowledge Based Systems of general purpose, such as the CommonKADS methodology (Common Knowledge Acquisition and Documentation Structuring) [27]. On the basis of the CommonKADS, MOKA was created and is now commonly used in engineering systems [28]. MOKA, being a result of interdisciplinary work of scientific teams and representatives of aeronautical and automotive industry, was prepared to achieve the following main goals [28]:

- reduction of risk, time and cost of development of KBE applications by 20–25%,
- ensuring development and maintenance of KBE applications,
- preparation of tools supporting application of the method,
- introduction of international KBE standard.

This paper proposes to use the basics of the MOKA as a fundamental methodology of building of VR applications for engineering, effectively making the VR apps the Knowledge Based Engineering solutions.

Methodology of building KB VR applications

Creators of the MOKA divided the work of building a KBE system into six distinct stages:

- identification – determination of purpose and range of building of the KBE system,

- justification – approximation of resources, costs and business risk,
- acquisition – gathering the knowledge from selected sources,
- formalization – formal recording of the gathered knowledge,
- application – implementation of knowledge in computer software,
- implementation – launching of the KBE system [28].

In the beginning, the aim of building the system needs to be determined, available resources must be estimated, a plan of the project should be prepared and the need of its building must be justified. Next, out of the indicated sources, the knowledge must be collected and recorded in a way to ensure possibility of its implementation in the chosen computer software. Stages related to identification and justification apply to organizational and economical aspects of building a KBE system, and in reality are omitted in MOKA, as are the two final stages – application and implementation – which concern development of an application, its installation, use and testing. MOKA focuses on stages of acquisition and formalization, which apply to methods of collecting and converting knowledge into language understandable for a computer application.

The author of this paper proposes to use the six stages determined in MOKA and, similarly as in MOKA, the most important stages are knowledge gathering and formalization (Fig. 4). The implementation is also very important to realize properly, as certain knowledge import mechanisms must be created (see Fig. 3). The next part of the paper describes these stages in greater detail.

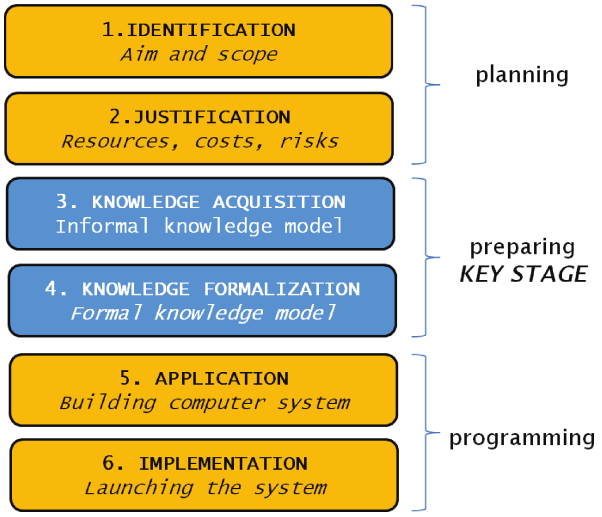

Fig. 4. Scheme of proposed methodology of building VR applications.

Identification and justification

Every VR application starts with definition of need (expectations) and purpose. At the conceptual stage, two basic questions need to be answered:

- Who will be using the application? Possible answers: shop floor workers, engineers, salesmen, inexperienced clients, professional clients etc.
- What is the purpose of the application? Possible answers: obtain new basic knowledge, obtain specialized knowledge and/or basic skills, obtain new knowledge and advanced practical skills.

On the basis of the answers to the questions above, an application can be further classified to one of three levels of knowledge presentation, as shown in Fig. 1.

Each level of applications has its own set of features and requirements, such as necessary types of data, means of interaction and expected behavior of presented virtual objects. These features correspond with specific software and hardware components of a VR system and allow to make a decision on how to build it, which is a problem tackled in many other elaborations [24, 29], so it will be not covered in greater details here.

Informal and formal knowledge acquisition

Knowledge acquisition and formalization is a key stage, on which the MOKA is focused [28]. In the concept presented by the author it is also the most important stage of preparation of a VR application. In general, knowledge about the product or the process can be stored in the following forms:

- text (descriptions, definitions, labels, numerical data etc.),
- 2D graphics (schemes, drawings, photographs, diagrams, illustrations, infographics etc.),
- audio (sounds of operation, notifications, speech),
- 3D models (CAD models and artist models),
- animations (often contained inside 3D models, sometimes universal – change of position of particular objects over time)

Apart from the knowledge mentioned above, the following logical relations between objects (depending on application type) should be defined:

- 3D and 2D object hierarchy and joints,
- definition of a sequence of operation with object links (what user will do, step by step, and what will be visible during realization of each step – for instance, in configurators – screens of configuration, in learning apps – tasks and their contents),
- definition of logical connections between objects (in configurators – options, values and logical links, in learning apps – physics and mutual behaviors of objects at a given time moment).

The basic knowledge must be gathered prior to starting any programming work in VR. Informal knowledge is gathered "as is" – it means getting all the necessary knowledge from the basic knowledge sources, such as databases of the company and interviews with its employees. At this stage, certain forms can be used, such as ICARE (Illustration, Constraints, Activities, Rules, Entities) [28–30], it may help organizing the knowledge.

In a traditional approach to making the VR applications, the informal knowledge is then hard coded inside the application, with no access to it from the outside after the application is released. Most programming environments for making the VR applications (such as Unity 3D or EON Studio) have in-built import mechanisms, for conversion of the standard image, text or audio files. Building the application (releasing it for a target platform) means encapsulating all the files to a single file package, after that operation all external access to knowledge within is lost, unless a VR developer decides to go back to the source code.

Formalization is the crucial stage in the proposed methodology, as its proper realization ensures flexibility of the created VR application and open access to the stored knowledge even after deployment of the app on a target platform, in a desired work space. The knowledge formalized by the guidelines of the proposed methodology should have a form understandable both by the computer algorithm and a human being. It should be also stored in a way enabling its future expansion outside the programming environment. In practice, in authors' experience, it means preparing a library of files containing particular knowledge entities, with a defined convention of naming, properties and internal structure of these files. Figure 5 presents an example of informal and formal storing of knowledge in an educational application (definition of a particular block in a theoretical, interactive 3D lesson). The informal knowledge is stored in forms, containing all the data types, complemented by specific files. The formal knowledge is stored in a text file of specific name, of specific structure, the other files (image, audio, 3D) also have specific naming convention and location.

It is worth mentioning, that in the proposed approach, the knowledge can be formally prepared in a total or partial way. In particular, it is usually very difficult to make 3D models available for edit outside the VR programming environment (not many commercially available software allows it directly). There is also a problem when complex logic is involved – formalization then usually means storing it in form of diagrams, using languages such as the UML (Universal Markup Language), which helps in building the application, but does not allow to edit the logic from the outside.

BLOK 1

TP (tekst pisany)

Wózki jezdniowe z napędem silnikowym są środkami transportu o ruchu pr
ograniczonym zasięgu, służące do poziomego jak i pionowego przemieszcza

TM (tekst mówiony)

Z pewnością wiesz jak wygląda wózek jezdniowy i do czego służy, jednak pc
nieskomplikowanej definicji. Wózki jezdniowe z napędem silnikowym są śrc
przerywanym, o ograniczonym zasięgu, służące do poziomego jak i pionowe
ładunków.

Według polskiej normy "PN-77/M-78100" dzieli się je ze względu na kryteri
elektryczny, benzynowy), cech eksploatacyjnych (np. podnośnikowy, ciągnił
kierowania (np. prowadzony, wyposażony w fotel).

G (grafika)

Podział wózków jezdniowych napędzanych wg polskiej normy "PN-77/M-78100"

Lp.	Kryterium	Nazwa podziału	Wyszczególnienie
1	Napęd	Rodzaj	Elektryczny(sieciowy); Akumulatorowy; Benzynowy; Diesel; Inny
2	Cecha eksploatacyjna	Typ	Naładowny; Unoszący; Podnośnikowy; Ciągnikowy; Specjalny
3	Sposób kierowania	Postać	Prowadzony; Podestowy; Wyposażony w fotel; Zdalnie sterowany

M (model z animacją)

Model: 3 wózki obok siebie (po jednym przykładzie na każde kryterium)

Bez animacji, po spojrzeniu na wózek – podświetlenie i wyświetlenie typu

C1_L1_B1.txt — Notatnik

Plik Edycja Format Widok Pomoc

```
B1;Podział wózków jezdniowych;
Wozek|0,0,0,0,0,0,1,1,1;
Wózki jezdniowe z napędem silnikowym są środkami transportu o
ruchu przerywanym, o ograniczonym zasięgu. Służą do poziomego
jak i pionowego przemieszczana ładunków. Według polskiej normy
dzieli się je ze względu na kryterium: napędu, cech
eksploatacyjnych i sposobu kierowania.
Zapoznaj się z przykładowymi wózkami jezdniowymi - podejdź i
obejrzyj je z bliska.
```

Fig. 5. Informal (top) versus formal (bottom) knowledge recording example.

That is why, in many cases, the formalization regards only application from level 1, sometimes 2 (see Fig. 1) and often the knowledge-based approach can be successfully applied only to text, 2D and audio data. This limitation is dependent on programming possibilities of a specific VR software, in the first place, but also current necessities and time frame of the realized project – the more urgent is a demand of implementing the application (e.g. due to short product lifecycle or losses due to untrained staff), the lesser focus will be put into full formalization.

Implementation and application

The stages of development of the VR applications, after knowledge is gathered and formalized, should be as follows:

1) Selection and obtaining of specific software and hardware components.

2) Preparing a visualization with full user navigation – a so-called virtual walk.

3) Programming interaction and behaviors of objects.

4) Building a graphical user interface and ensuring final user communication methods.

5) Internal verification and publication for use by a sample of end users.

The stages are similar to stages that can be found in the process of building any VR application. Detailed guidelines of building realistic VR applications have been a subject of numerous research in previous years [1, 2, 5, 6, 19, 24, 29] and are beyond the scope of this paper. Only the most important stages of the process from the viewpoint of the knowledge-based approach – 2 and 3 – will be mentioned here.

Traditionally, preparing the visualization means importing 3D geometry, assigning materials and textures and composing the scene with lighting, shadows and reflections. In the full-fledged knowledge-based approach, the virtual scene is empty at the runtime beginning (except some static, constant elements, characteristic for a given application) and all the 3D and 2D components are instantiated during the run-time, on the basis of formalized knowledge about how the current scene should look like. The instantiation means run-time loading of 3D data, assigning materials and textures to it and placing it inside the scene at a given position, orientation and with a given scale. In the traditional approach, for instance, building a configurator of 30 vehicles requires building 30 separate scenes, with hard-coded visualization. In the proposed, new approach, it requires building only one scene, with a logic for run-time object instantiation. This is much more flexible in terms of further expanding of any VR application, but also has certain limitations – for example, using so-called prebaked lighting is not fully possible, it also makes the application more demanding in terms of computing power.

In terms of the stage 3 – the object behavior programming – this process again can benefit from knowledge formalization. Certain logical connections between objects can be recorded to external text files (or schemes, UML diagrams etc.) and then read during the runtime, building logic dynamically. An example may be a logic of options in a configurator – for example, exclusions and enforcements (i.e. what options must be switched on or off when another option is selected) can be stored externally and modified at will, without need of getting back to the code when a need of logic changing arises.

Methodology summary

It is worth noting, that building VR applications according to the described methodology is much more difficult and, initially, time consuming than the traditional process. It requires a number of considerations and decisions – which parts of a certain product/process knowledge should be formalized and which should not, what shape should a knowledge library have, what should be the file structure etc. As the approach is new, there are no strict guidelines here – they will probably be created in the upcoming years, as more of such solutions will be created and implemented.

One important consideration is that a VR application built with a full knowledge-based approach is no longer a typical, traditionally understood VR app, but becomes a framework Knowledge Based System instead. See examples below – the Virtual Design Studio was created as a city bus configurator, but its flexible structure allows it to be any product configurator, so it became a framework solution for building interactive VR configurators. Similarly with the Virtual Skill Teacher – it can be used to build any type of educational applications for technical skills. After initial building of the application, adding more content is no longer performed in the programming environment, meaning that it can be done by virtually anyone inside the company. This makes this approach very powerful and allows fulfilling requirements towards manageability, stated by the production companies.

Examples of KB VR applications

Virtual Design Studio

The Virtual Design Studio (VDS) is a framework solution for VR configuration of vehicles. This tool is based on the EON Studio software and it aids communication between a company and a client, allowing to avoid problems and mistakes related to selection of product variants in a traditional way, without dynamic visualization on each stage of feature selection. The VDS system allows storing visual and logical, properly structured knowledge about a given product (vehicle), as well as its processing, sharing and supplementing by the system users in order to aid the engineering activities in a company. On the other side, it is a typical "VR in design" solution, allowing interaction in 3D with a product model, placed in a specific environment. The system allows, among other things, to fully configure visual features of a vehicle (more than 100 options), display multimedia and technical information (explore a given vehicle),

as well as easily export and import saved variants, for accelerated generation of product documentation and archiving contracts.

The system allows work in two stages (Fig. 6) – the main configuration is performed using a touch screen interface and large-screen 3D projection, while the immersive configuration is performed in an HMD and allows virtual walk and inspection of a created variant (Fig. 7). Both stages are synchronized in real time.

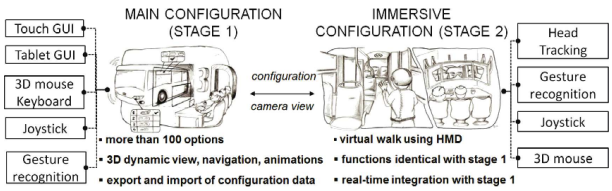

Fig. 6. Scheme of Virtual Design Studio.

Fig. 7. Working in the Virtual Design Studio.

The knowledge in the system is fully formalized, and creation of the system was a premise of preparing the methodology presented in this paper. Almost anything regarding the contents of the system can be changed from the outside of the programming environment, using a special, easy management tool created by the authors. Thus, the application can be freely expanded without need of any programming whatsoever. However, there are plenty of knowledge types that need to be prepared, starting from 3D models, materials, textures etc., through configuration scope definition, to logic of the whole configuration process.

The initial build of the system took 2 years of work of a 7-person team of programmers, engineers and graphics designers and it contained fully configurable models of two vehicles. The system was implemented in an industrial company – a large city bus manufacturer and it is successfully used there. It has been tested by the author on two separate cases, that adding another, completely different product to the system, with full configuration and presentation features, is a work of 3 months for a single, inexperienced engineer. The massive reduction of time of creating another "configurator" type application, comparing to the initial build, is achieved thanks to the knowledge-based approach introduced by the team of creators of the system.

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:

1. the course is the main entity, it is divided into lessons and exercises, which are further divided into blocks,
2. lessons are theoretical and contain a number of graphical, text, audio and interactive 3D info presented in immersive space (Fig. 8),
3. exercises are sets of practical tasks – they require certain actions in a specified time in order to be passed,
4. each course has a set of questions, the exam mode is in-built and also performed in immersive space.

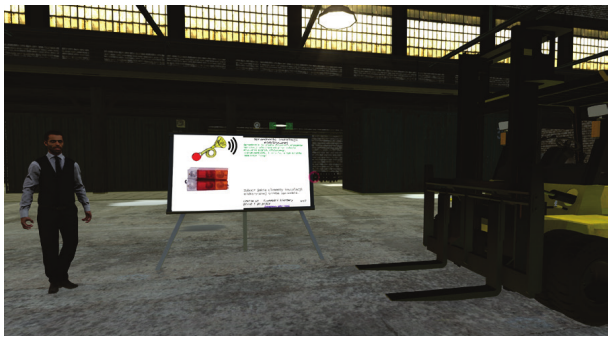

Fig. 8. Virtual Skill Teacher – lesson mode.

The knowledge stored in the application is formalized and the whole platform was created according to the methodology presented in this paper. The text, audio and 2D data, as well as some minor logic (lose and win conditions in exercise blocks) can be freely edited and expanded from the outside of the programming software (Unity engine was used to build the whole platform). The interactive 3D models (so-called prefabs) are created inside the Unity platform, so the formalization, so far, is only partial. However, assigning prefabs to lesson or exercise blocks is also possible from the outside of the programming software. There is also a mechanism of composing pre-defined logic (certain behaviors, like "click-to-animate" or "click-to-show") with 3D models during the runtime, so theoretically any prefab

can be created just by a formal definition of logic behind it. However, 3D models cannot be loaded from the outside, so the authors will prepare a large library of sample models, for future edition and expansion of the available range of courses.

The Virtual Skill Teacher is still in the development phase, its first prototypes were implemented in a company performing industrial trainings. The knowledge-based approach was proven to be very effective – the first prototype of the system was programmed in 3 months by three VR programmers, containing 3 lessons and 1 exercise. Adding another 6 lessons and 3 exercises took less than 3 weeks of work of only one VR programmer. The vast time reduction was achieved thanks to formalization and knowledge import mechanisms – the only task to perform while expanding the lesson library was to prepare the knowledge in an appropriate format.

Conclusions

The new approach to building VR applications for industry, presented in the paper, was proven to be effective in practice. It requires changing some of traditional views on particular stages of development of VR applications, but in exchange allows building flexible, manageable and easily extendable solutions. These solutions can be effectively implemented in production companies, with a decreased risk of discontinued use of application after initial period of implementation, which is a frequent case regarding traditionally built applications with hard-coded knowledge.

The proposed knowledge-based approach requires significantly more work in the programming stage, so it is advisable to put the presented ideas into practice only when the requirements towards applications directly indicate a need of further vast knowledge base expansion during a prolonged time after initial launch of the application. If the product is short-lived and only a simple visualization is required, or a certain process to be simulated remains relatively unchanged for a number of years, this approach may unnecessarily delay initial implementation and thus reduce benefits of use of the Virtual Reality technology. 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 and in what context.

To sum up, it can be stated that use of Knowledge Engineering methodologies such as the MOKA can be justified in building of any computer system for engineering purpose, regardless of the technology.

Development of one of the presented case studies was financed by Polish National Centre for Research and Development, INNOTECH program. Another case study development was financed indirectly by Operational Program Knowledge Education Development, realized by Polish Ministry of Development.

References

[1] Scherman W.R., Craig A.B., *Understanding virtual reality: interface, application, and design*, Morgan Kaufmann 2003.

[2] Burdea G.C., Coiffet P., *Virtual reality technology*, John Wiley & Sons, Inc., 2003.

[3] Bowman D.A., McMahan R.P., *Virtual reality: how much immersion is enough?*, Computer, 40, 7, 36–43, 2007.

[4] Jezernik A., Hren G., *A solution to integrate computer-aided design (CAD) and virtual reality (VR) databases in design and manufacturing processes*, International Journal of Advanced Manufacturing Technology, 22, 11, 768–774, 2003.

[5] Martin G., Wolfgang S., Ralph S., Kathrin B., *Metamodel for VR-based design reviews*, Proceedings of the International Conference on Engineering Design, ICED, 4, DS87-4, 337–346, 2017.

[6] Colombo S., Nazir S., Manca D., *Immersive virtual reality for training and decision making: preliminary results of experiments performed with a plant simulator*, SPE Economics & Management, 6, 4, 2014.

[7] Grajewski D., Górski F., Zawadzki P., Hamrol A., *Application of virtual reality techniques in design of ergonomic manufacturing workplaces*, Procedia Computer Science, 25, 289–301, 2013.

[8] Mujber T.S., Szecsi T., Hashmi M.S.J., *Virtual reality applications in manufacturing process simulation*, Journal of Materials Processing Technology, 155, 1834–1838, 2004.

[9] Jiang M., *Virtual reality boosting automotive development*, Virtual Reality & Augmented Reality in Industry, Springer Berlin Heidelberg, 171–180, 2011.

[10] Buń P., Górski F., Wichniarek R., Kuczko W., Zawadzki P., Hamrol A., *Application of professional and low-cost Head Mounted Devices in immersive educational application*, Procedia Computer Science, 75, 173–181, 2015.

[11] Escobar-Castillejos D., Noguez J., Neri L., Magana A., Benes B.. *A review of simulators with haptic devices for medical training*, Journal of Medical Systems, 40(4), 104, 2016, doi: 10.1007/s10916-016-0459-8.

[12] Martín-Gutiérrez J., Mora C.E, Añorbe-Díaz B., González-Marrero A., *Virtual technologies trends in education*, EURASIA Journal of Mathematics, Science and Technology Education, 13(2), 469–486, 2017.

[13] Zawadzki P., Żywicki K., *Smart product design and production control for effective mass customization in the Industry 4.0 concept*, Management and Production Engineering Review, 7(3), 105–112, 2016.

[14] Żywicki K., Zawadzki P., Górski F., *Virtual reality production training system in the scope of intelligent factory*, Advances in Intelligent Systems and Computing, Springer, 637, 450–458, 2018.

[15] Ye J., Badiyani S., Raja V., Schlegel T., *Applications of virtual reality in product design evaluation*, HCI'07 Proceedings of the 12th International Conference on Human-Computer Interaction: Applications and Services, 4553 (Part 4), 1190–1199, 2007.

[16] Abulrub A.H., Attridge A., Williams M.A., *Virtual reality in engineering education: the future of creative learning*, International Journal of Emerging Technologies in Learning, 6(4), 4–11, 2011.

[17] Rogers S., *Technical advances in the automotive industry*, retrieved from: retail-focus.co.uk, 2016.

[18] Górski F., Buń P., Wichniarek R., Zawadzki P., Hamrol A., *Immersive city bus configuration system for marketing and sales education*, Procedia Computer Science, 75, 137–146, 2015.

[19] Yan M., Chen X., Zhou J., *An interactive system for eficient 3D furniture arrangement*, ACM International Conference Proceeding Series, Volume Part F128640, 27 June 2017, Article number a29.

[20] Rodriguez J., Gutiérrez T., Sánchez E.J., Casado S., Aguinaga I., *Training of procedural tasks through the use of virtual reality and direct aids*, Virtual Reality and Environments, 2012.

[21] Wan H., Gao S., Peng Q., Dai G., Zhang F., *Mivas: a multi-modal immersive virtual assembly system*, Proceedings of DETC'04, 2004.

[22] Grajewski D., Górski F., Zawadzki P., Hamrol A., *Immersive and haptic educational simulations of assembly workplace conditions*, Procedia Computer Science, 75, 359–368, 2015.

[23] Harley D., Tarun A.P., Germinario D., Mazalek A., *Tangible VR: diegetic tangible objects for virtual reality narratives*, DIS 2017 Conference Paper, pp. 1253–1263, 2017, doi: 10.1145/3064663.3064680.

[24] Buń P., Górski F., Zawadzki P., Wichniarek R., Hamrol A., *Selection of optimal software for immersive virtual reality application of city bus configurator*, Advances in Intelligent Systems and Computing, 571, 480–489, 2017.

[25] Develop3D.com, *Review: VRED Pro 2017 & VR*, retrieved from: http://www.develop3d.com/reviews/vred-pro-2017-vr-visualisation-design-rendering, 2017.

[26] Tiwari V., Jain P.K., Tandon P., *Design process automation KBE*, Proceedings of the World Congress on Engineering, vol. II, 2013, retrieved from http://www.iaeng.org/publication/WCE2013/WCE2013_pp737-742.pdf.

[27] Schreiber G., *Knowledge engineering*, [in:] Handbook of Knowledge Representation, Lifschitz V., van Harmelen F., Porter B. [Eds.], Elsevier Science, pp. 929–946, Amsterdam 2007.

[28] Stokes M., *Managing engineering knowledge: MOKA methodology for knowledge based engineering applications*, Professional Engineering Publishing, London 2001.

[29] Górski F., Bun P., Wichniarek R., Zawadzki P., Hamrol A., *Effective design of educational virtual reality applications for medicine using knowledge-engineering techniques*, EURASIA Journal of Mathematics, Science and Technology Education, 13, 2, 395–416, 2017.

[30] Górski F., Zawadzki P., Hamrol A., *Knowledge based engineering as a condition of effective mass production of configurable products by design automation*, Journal of Machine Engineering, 16, 4, 5–30, 2016.