



Construction and Implementation of Metaverse Intelligent Classroom Based on Domain-specific Language

Yingmei Wei^{1,a*}, Yu Li^{1,b}, Yuxiang Xie^c

College of Systems Engineering, National University of Defense Technology, Changsha, China

¹These authors contributed equally to this work and should be considered co-first authors

^{a*}weiyingmei@nudt.edu.cn, ^bliyu23e@nudt.edu.cn

^cyxxie@nudt.edu.cn

Abstract. In this paper, a metaverse intelligent classroom design based on Domain-specific Language is proposed to address the time, space and form constraints of traditional teaching methods, as well as the problems of insufficient interaction and inconspicuous personalised teaching in existing virtual teaching systems. A general framework based on Unity and Spring Boot is built through the collaborative application of tools such as Blender, Unreal, Visual Studio and IntelliJ IDEA. A lightweight compiler was implemented and a generic syntax was designed to make it easy to use for users without programming experience. Future research can expand virtual courses and experiments in more subject areas, combining augmented reality, artificial intelligence and other technologies to further enhance the usefulness and richness of the virtual teaching system, so as to meet the learning needs of different students and to improve learning effectiveness and satisfaction.

Keywords: Metaverse; Virtual Reality; Domain-specific Language; Unity; SpringBoot.

1 Introduction

The development of metaverse technology can be traced back to the 1990s of the twentieth century. With the continuous improvement of computer technology and graphic processing capabilities, virtual reality technology has gradually advanced, laying the foundation for the development of the metaverse.

In recent years, with the development of blockchain [1], big data [2], artificial intelligence [3] and cloud computing technology [4], metaverse technology has received more support and development. Blockchain technology provides the advantages of decentralisation, security and verifiability for metaverse, which makes the transactions and asset management of metaverse more secure and transparent. Massive amounts of data need to be processed in the metaverse, and big data technology can

© The Author(s) 2024

M. Yu et al. (eds.), *Proceedings of the 2024 5th International Conference on Big Data and Informatization Education (ICBDIE 2024)*, Advances in Intelligent Systems Research 182,

https://doi.org/10.2991/978-94-6463-417-4_12

provide efficient data storage, management, processing and analysis capabilities, helping the metaverse achieve high-performance data processing and analysis. Artificial intelligence technology provides the metaverse with more intelligent user experiences and more complex virtual world scenarios, such as more realistic natural language processing and emotion recognition. Cloud computing technology provides the metaverse with a more stable and efficient infrastructure, enabling smoother and more immersive user interactions. The metaverse is the foundation for engaging in online activities in the future and the future development of virtual reality technology [5].

In the information age, digitalisation has become a key and indispensable factor in education. With the continuous emergence of information technology and the acceleration of the technological revolution, the traditional classroom teaching system is facing unprecedented challenges and changes. The process of education informatisation is evolving rapidly, and digital education has become a driving force for the transformation of learning styles and teaching modes [6].

The traditional classroom model has some significant drawbacks. Firstly, its rigid teaching structure restricts the cultivation of students' creative thinking and practical abilities, as the curriculum tends to stick to a set teaching method. Secondly, personalised learning needs are not adequately met, as the classroom is difficult to adapt to the learning pace and interests of different students. In addition, there are limited opportunities for interaction and students lack active participation and co-operation in the traditional mode. Finally, traditional teaching materials are not updated in a timely manner to reflect the latest knowledge and developments, resulting in students being exposed to information that may be outdated. These problems have led to a gradual challenge to the traditional classroom model, and the field of education is seeking more flexible and adaptable teaching methods. The pace of change in information technology is much faster than the pace of development of classroom teaching, which poses a great challenge to the traditional mode of teaching and learning, and therefore there is an urgent need to reform classroom teaching and learning.

The technological revolution drives educational change, and metaverse empowerment education provides a new direction for classroom teaching informatisation in the new era. With the integration of metaverse elements in classroom teaching, its development space will be redefined, and this teaching mode will also better meet the individual needs of students and promote their overall development. [7]

Metaverse classrooms based on Domain-specific Language (DSL) are still in the exploratory and developmental stage and are not yet widely used in education. In fact, it is important to customise interactive exploration languages. DSL can help build more personalised and flexible educational content that better meets the learning needs and interests of different students. At the same time, DSL can also help build more realistic virtual worlds to improve students' engagement and experience.

This paper implements a metaverse classroom construction based on DSL. Three courses, biology, geography and chemistry, are set up as demonstrations based on different teaching contexts. 3D modelling, online video communication, online message interaction, and voice explanation of knowledge points are used to simulate a real classroom, in order to enhance the immersion of teaching and learning, and to increase the attractiveness and richness of the teaching content. We implemented a

DSL interactive scripting tool and designed the script syntax according to the interaction requirements. Non-professional users who lack programming experience can easily master the method of building virtual classrooms in a short period of time after training. Users can select, customise and combine virtual resources, and flexibly customise the classroom display content to meet the educational needs of different fields. The system adopts a client/server (C/S) architecture, which allows users to interact with the platform through web pages on the browser side.

2 Related Work

The development of virtual reality technology can be traced back to the 1960s [8]. The United States and other places pioneered the emergence of early VR technology products. In 1962, Morton Heilig of the United States developed a street simulator [9]. In 1965, Ivan Sutherland, director of the Office of Information Processing Technology at the United States Advanced Research Projects Agency, put forward the idea of transforming a computer display into a window on the objective world in his paper *The Ultimate Display*, and successfully developed a helmet-mounted graphic display. This is considered a seminal moment in virtual reality technology research [10].

However, it was not until the 17th International Conference on Graphics in 1990 that virtual reality technology received widespread attention [11]. Graphics experts from around the world reported on recent advances in input/output device technology and multimedia technologies such as audio and video, which have pushed the transition of interactive technology from two-dimensional to three-dimensional and even multidimensional directions. In the area of three-dimensional interactive technology, a more mature virtual reality system was demonstrated at the International Graphics Conference in 1992, marking substantial progress in this technical field.

The Department of Computer Science at the University of North Carolina (UNC) is one of the earliest and most prominent universities in the world for virtual reality research [12]. The Department's research in VR covers a number of important directions, including molecular modelling, airline piloting, surgical simulation, and architectural simulation, among other areas.

The Massachusetts Institute of Technology (MIT) is a pioneer in the fields of artificial intelligence, robotics, computer graphics, and animation [13]. These technologies largely form the basis of virtual reality (VR) technology. In 1985, MIT established the MIT Media Lab to begin formal research on virtual environments. Media Lab is one of the world's leading research and academic organisations, combining a vision of the digital future with a new style of creative invention dedicated to achieving an interdisciplinary fusion of technology, media, art and design.

The Human Interface Technology Lab (HIT Lab) at the University of Washington's Washington Technology Centre is bringing VR research to education, design, entertainment and manufacturing [14]. Virtual Retinal Display (VRD) technology was led by Professor Tom Furness of the HIT Lab in the early 1990s. This headset projects low-power lasers directly onto the retina to add virtual objects to the real world. Today's Magic Leap's light field borrows many techniques and ideas from it.

In international contexts, the research and implementation of metaverse technology have reached a relatively advanced stage, marked by the successive introduction of related technologies and products. However, the exploration of virtual classrooms is still in its nascent phase presently [15]. Horizon Workrooms [16], an application tailored for office scenarios within Facebook's Horizon social platform, constitutes a noteworthy advancement. This service facilitates users in conducting meetings within a virtual reality milieu, elevating immersion through the utilization of devices like head-mounted displays. Distinguished as Facebook's inaugural software with a focus on gesture recognition, Horizon Workrooms stands as a pioneering application in augmenting functionality within virtual environments. Concurrently, each virtual room accommodates up to 16 users, ensuring simultaneous online engagement. For individuals devoid of head-mounted displays, participation remains accessible through video calls. Horizon Workrooms encompasses various additional features, encompassing desktop and keyboard recognition, avatars, mixed reality, and gesture tracking. The cohesive integration of these features not only enriches but also amplifies the interactive and immersive nature of the virtual collaboration experience for users.

Compared with the progress in the field of VR technology in countries such as the United States and the United Kingdom, which have already developed maturely, there is a certain gap in the level of development of VR technology in China due to a variety of factors. However, in recent years, the country has begun to gradually focus on the application of VR technology in the field of education and actively support relevant research programmes to bridge the technology gap in this field.

3 System Architecture

In the system architecture, we have clarified the functions and interrelationships of each key component. The design of the system architecture not only focuses on meeting teaching and learning needs, but also considers the performance, maintainability and scalability of the system. By integrating advanced virtual technology and DSL, we expect to build a metaverse intelligent classroom that can meet the needs of teaching and learning.

3.1 General System Structure

The system's architecture is illustrated in Fig. 1 It is composed of three essential components: the client side, the server side, and the background account management platform. The client side comprises two modules: the classroom building module and the smart classroom module. The former empowers educators to create and customize virtual classrooms, while the latter provides students with an environment conducive to self-directed exploration and learning. The backend management platform comprises three modules with distinct permissions, allowing the Academic Affairs Office to manage user accounts. The developer utilizes the server side to process runtime

data from the client side and the account management platform, ensuring the comprehensive maintenance of the entire system.

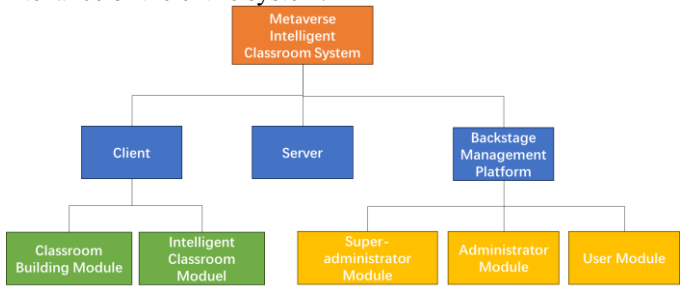


Fig. 1. General System Structure

3.2 System Technical Structure

Client-side technical architecture: The intelligent classroom teaching system adopts the client/server (C/S) architecture, in which the client side uses Unity and Visual Studio as the building tools of the development environment. The server side mainly implements the information exchange function between the Unity side and the database. The core functions of the system, including user registration and login, intelligent teaching aids and DSL interaction scripts for rapid scene layout, are mainly realised by the service layer. For data interaction, the DAO layer adopts Mybatis-Plus and MySQL technology, and MySQL is mainly used to store students' basic information. The client connects to the MySQL database by importing the MySqlConnection module to interact with the account information. With this combination of technologies, this virtual teaching system is able to provide efficient and stable services to provide a quality virtual learning experience for student users. Fig. 2 depicts the technical architecture of the system client.

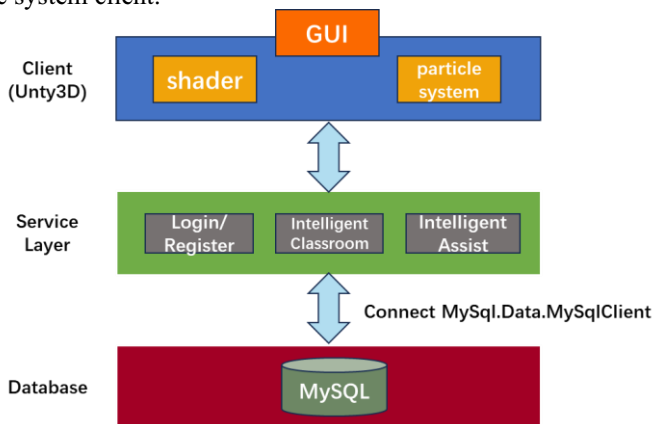


Fig. 2. System Client Technical Structure

Back-stage account management platform technical architecture: The platform is developed using IntelliJ IDEA, and user interaction is facilitated through web pages on the browser side. The control layer manages HTTP requests, encompassing GET, POST, PUT, DELETE, and executes functions related to addition, deletion, retrieval, and modification. The service layer interfaces with the DAO layer, implementing relevant functionalities. The DAO layer utilizes Mybatis-Plus and MySQL for data interaction. The technical architecture of the backend account management platform is illustrated in Fig. 3

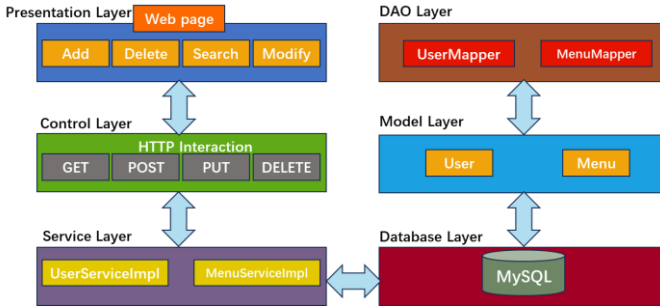


Fig. 3. System Back-stage Account Management Platform Structure

4 System Design

In this section we delve into the system design of a metaverse intelligent classroom based on DSL. Through the system design, we seek to create a flexible, interactive and personalised virtual educational environment that caters for different disciplines and learning needs.

4.1 Virtual Environment Design

Virtual Teaching Material: The virtual classroom plays a pivotal role in the Metaverse Intelligent Classroom. Through the adoption of an immersive virtual classroom design, students are transported as if into a physical classroom. Within this virtual environment, we offer a wealth of three-dimensional teaching materials for educators to flexibly utilize, as shown in Fig. 4, while also supporting the upload of educators' own teaching materials. Moreover, the construction of the virtual classroom possesses a high degree of flexibility, allowing customization to meet the requirements of different subjects and educational needs. Taking a geography class as an example, educators have the capability to transform the virtual classroom into cosmic space, enabling students to delve into the composition of the solar system. This personalized and flexible design renders the virtual classroom a rich and captivating learning space.

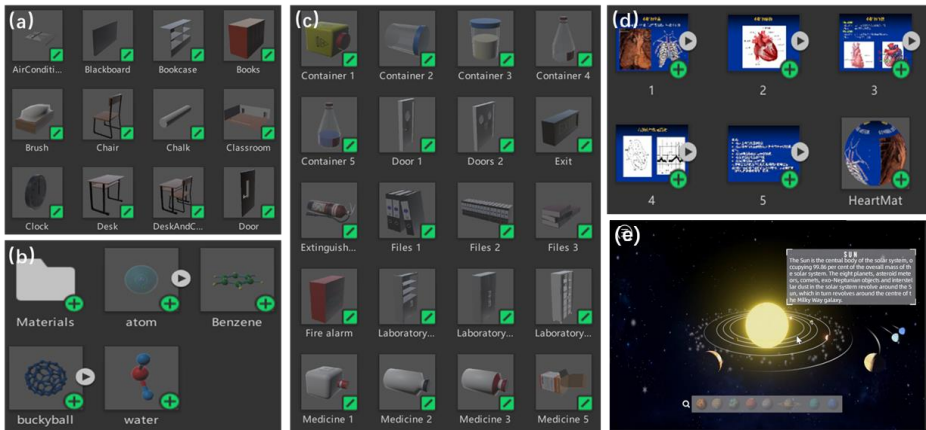


Fig. 4. Virtual Teaching Materials

Virtual Character: In the virtual environment, role-playing plays a crucial role in teaching. As shown in Fig. 5, we have created virtual roles for each student, enabling them to interact and learn within the virtual environment. These roles are not merely virtual representations but can also interact based on students' instructions, simulating a real-world learning experience. Additionally, we have designed virtual roles for educators to guide students in the virtual environment, providing guidance and support.

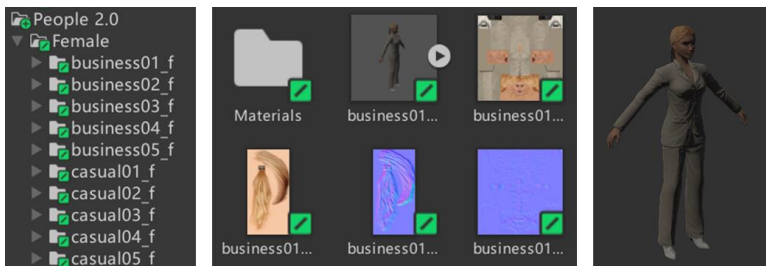


Fig. 5. Virtual Character

4.2 System Function Design

UI Interaction: The design of the UI interaction module prioritizes user-friendliness to ensure seamless interaction between educators and students within the system. This module supports multiple input modes, including mouse, keyboard, and voice recognition, allowing users to interact based on their individual preferences and needs. For instance, students can control virtual experiments using a mouse, while educators can provide guidance through voice commands. The UI interaction module enables users

to manipulate objects and elements within the virtual environment, performing operations such as selection, movement, rotation, and modification of properties.

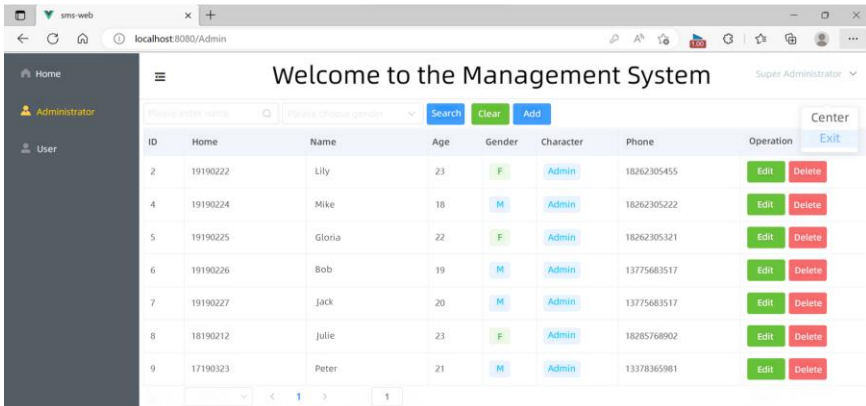
Intelligent Speech Recognition: Speech recognition is particularly crucial in the VR domain as it not only simulates AI conversations with users but also provides more options for users to communicate with any application. If an application has too many buttons or other GUI elements, it can quickly overwhelm users. However, with voice control, various operations become easily accessible by speaking in the VR environment. In this project, specific commands are designed for elements in the metaverse classroom scenario, allowing users to personalize the arrangement of teaching scenes through voice commands.

Back-stage Management:

If the account management module is directly established within Unity, it would result in excessive redundancy of system functions. Therefore, this paper proposes managing accounts on the browser side. Additionally, managing accounts on the browser side is advantageous for administrators to view and modify account information.

Firstly, create a Spring Boot project through IDEA with project management using Maven, Java as the language, JAR as the packaging method, and JDK version 1.8. Add necessary dependencies for the project such as Spring Web, Lombok, MySQL, MyBatis-Plus, etc.

Fig. 6 depicts the display of the front-end page: the development framework used for constructing the front-end page is Vue, utilizing the Container layout container from element-ui to populate the webpage interface. JavaScript language is employed for front-end page development, filling data from the controller layer into the front end.



ID	Home	Name	Age	Gender	Character	Phone	Operation
2	19190222	Lily	23	F	Admin	18262305455	Edit Delete
4	19190224	Mike	18	M	Admin	18262305222	Edit Delete
5	19190225	Gloria	22	F	Admin	18262305321	Edit Delete
6	19190226	Bob	19	M	Admin	13775683517	Edit Delete
7	19190227	Jack	20	M	Admin	13775683517	Edit Delete
8	18190212	Julie	23	F	Admin	18265768902	Edit Delete
9	17190323	Peter	21	M	Admin	13378365981	Edit Delete

Fig. 6. Front-end Account Management Page

4.3 DSL Design

Design Goal: The DSL script interaction module designed in this project has the following design objectives:

- to ensure that the interactive interface is user-friendly, with clear operational logic, providing an excellent user experience;
- to support users in quickly creating and editing DSL for the rapid arrangement of teaching scenes;
- to have a certain level of security, preventing users from engaging in malicious activities and attacks using scripts, with error prompts for code that violates syntax rules;
- to allow users to save, load, and share customized scripts.

Initially, users interact by entering code through the script editing box. Mouse interaction allows users to freely change the size of the script editing box and hide it, ensuring a clear and user-friendly interactive interface.

Secondly, the script editing box is equipped with six interactive GUI buttons, enabling users to automatically generate corresponding code through mouse interaction. Users only need to input the corresponding data in the blank spaces. This project includes a lightweight compiler that can transform syntax statements resembling natural language into C# language.

Furthermore, this design incorporates a certain level of security. When the code is submitted for compilation, the compiler checks whether the syntax of the code is standard. If there are syntax errors in the language input by the user, the interface will automatically prompt an error warning. This helps users identify syntax errors in the code promptly, enhancing the reliability and quality of the code.

Lastly, this design includes a save function, enabling the storage of configured script languages. The backend will transform syntax statements resembling natural language into XML language for storage. If users need to arrange similar scenes in the future, they can simply parse and read the XML file, significantly improving configuration efficiency.

DSL Grammar Rule: The interactive syntax designed in this paper consists of four types: object loading, transformation, animation playback control, and camera (main viewpoint) positioning.

Object Loading Operation: The load statement design resembles an overloaded function in C++, supporting multiple data types. This convenience is beneficial for users unfamiliar with programming, allowing them to load various types of data.

Algorithm 1 Load Object

Input: *name*: object to be loaded; (x, y, z) : position, size or rotation;
Output: Execution succeed ? 1: 0
 initial *name* = *NULL* and $(x, y, z) = (0, 0, 0)$;
 load{
 gameObject(*name*);
 location(x, y, z);
 scale(x, y, z);
 rotation(x, y, z);
 }

Object Transformation Operation: Object transformation in this system is encapsulated in three identifiers—rotate, scale, and translate. The rotate identifier can be overloaded based on different parameters, allowing rotation along a specific coordinate or directly inputting a three-dimensional coordinate vector. The scale function can proportionally scale the object, while the translate function can move the object along a specified direction. This encapsulation enhances the flexibility and intuitiveness of object transformation, making it convenient for users to perform related interactive operations. The parallel code block can accommodate multiple basic interactions (rotation, translation, and scaling) simultaneously.

Algorithm 2 Rotate Object

Input: *name*: object to be rotated; *axis*: the axis of the object; *angle*: angle of rotation
Output: Execution succeed ? 1: 0
 initial *name* = *NULL*, *axis* = *NULL* and *angle* = 0;
 rotate{
 gameObject(*name*);
 axis(*axis*);
 angle(*angle*);
 }

Algorithm 3 Translate Object

Input: *name*: object to be translated; (x, y, z) : position to move;
Output: Execution succeed ? 1: 0
 initial *name* = *NULL* and $(x, y, z) = (0, 0, 0)$;
 translate{
 gameObject(*name*);
 to(x, y, z);
 }

Algorithm 4 Scale Object
<p>Input: <i>name</i>: object to be scaled; (x, y, z): size; Output: Execution succeed ? 1: 0 initial <i>name</i> = <i>NULL</i> and $(x, y, z) = (0, 0, 0)$; scale{ gameObject(<i>name</i>); factor(<i>x, y, z</i>); }</p>

Animation Control Operation: *The syntax for animation control manages the creation of data presentation animations, exemplified by the animate statement, which sets the global animation speed.*

Algorithm 5 Animation Object
<p>Input: <i>name</i>: animation; <i>time</i>: animation runtime; Output: Execution succeed ? 1: 0 initial <i>name</i> = <i>NULL</i> and <i>time</i> = 0; animation{ animation(<i>name</i>); speed(<i>time</i>); }</p>

Camera Positioning Operation: Manipulating the keyboard for camera movement may lead to challenges in observing an object from a specific viewpoint due to inadequate fine movement scales. The Camera Positioning Control syntax enables precise adjustment of the camera position, facilitating a more detailed observation of the instructional elements in the scene.

Algorithm 6 Camera Control
<p>Input: <i>location</i>: position of camera; <i>rotation</i>: rotation of camera; Output: Execution succeed ? 1: 0 initial <i>location</i> = $(0, 0, 0)$ and <i>rotation</i> = $(0, 0, 0)$; camera{ location(<i>location</i>); rotation (<i>rotation</i>); }</p>

5 System Implementation

This section will present the login interface, menu interface, geography lesson scene, biology lesson scene, chemistry lesson scene, DSL script control, and the front-end webpage of the student management system based on Spring Boot and Vue, respectively.

5.1 Login Interface

As shown in Fig. 7, the login interface of this system is presented. At the top of the interface, there is a welcome title, and in the middle, there are two input fields for "Account" and "Password," with the content of the "Password" input field displayed in encrypted form. Users click on the input fields to enter their account and password, and upon clicking the login button, the text content is retrieved and matched with the user account database entries. If the entered account and password match the database entries, indicating the correct account and password, the interface will navigate to the menu interface. Otherwise, an "Error" prompt box will appear on the interface.

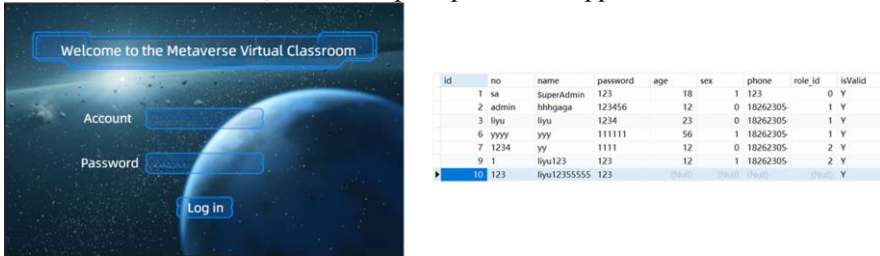


Fig. 7. Login interface and account information in the database

5.2 Menu Interface

As illustrated in Fig. 8, this system's menu interface is presented. The menu interface is equipped with four options: "Geography Class," "Biology Class," "Chemistry Class," and "Exit." Selecting the first three options allows entry into the corresponding classroom scenes. Clicking on the "Exit" option immediately exits the program. The menu interface utilizes features such as particle systems and Animator to achieve dynamic effects. For instance, when the mouse hovers over or clicks on an option button, the button's color undergoes subtle changes, providing users with a clearer indication of the option they are interacting with.

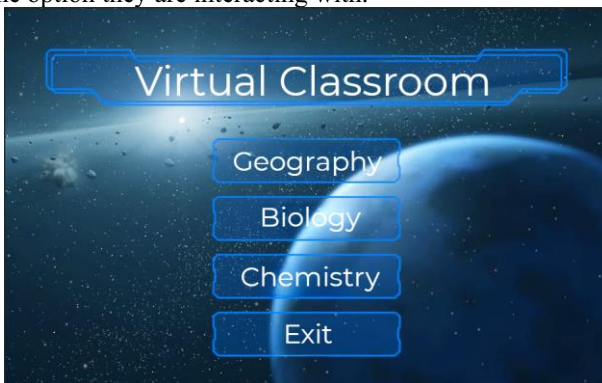


Fig. 8. Menu Interface

5.3 Teaching Scene

Geography Class Interface:

As shown in Fig. 9, this is the main interface of the geography lesson. Clicking the button in the upper left corner can return to the main menu, and the dropdown list in the upper right corner can switch teaching elements. The system is designed with three options: "Earth Structure," "Contour Topographic Map," and "Solar System."



Fig. 9. Geography class main interface

As shown in Fig. 10, clicking on the "Earth Structure" option reveals a three-dimensional cross-section of the Earth in the scene. Users can use mouse clicks and drag to carefully observe and learn about the internal structure of the Earth. Clicking on different parts of the Earth – the core, outer core, mantle, and crust – will display detailed knowledge points on the interface. Clicking on the teacher will provide synchronized voice explanations. Additionally, the presentation screen simulates displaying slides and videos. Clicking on the mouse can enlarge the display of slides and videos on the UI interface. After clicking on the slides, the arrows on both sides of the interface can be used to switch between different pages of the slides. In the lower-left corner of the interface, there are three icons for controlling video playback, from left to right: start, pause, stop. Moreover, the system includes voice recognition functionality, allowing users to control the display and disappearance of elements through voice commands such as "open/play slides" and "open/play/pause/stop video."

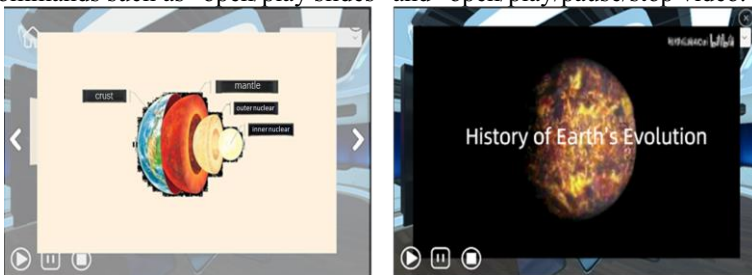


Fig. 10. Mouse click or voice recognition for PPT and video zoom display

Selecting the "Solar System" option transitions to another scene. In this scene, 3D modeling is performed using Blender, including texture mapping, self-illumination setup, particle system application, and the addition of C# script components. This implementation achieves realistic and vivid rotations of the planets around the sun in the solar system. Dragging the mouse on the interface allows for changing the rotation angle of the solar system, facilitating observation from different perspectives. Clicking on specific planets will display detailed information about the planet in the upper-right corner of the UI interface. Simultaneously, the camera switches from an overall view to tracking the current planet, making it convenient for detailed observation and learning. This experience allows users to understand the relative sizes and orbital speeds of the planets. Clicking on the icon in the upper-left corner resets the observation perspective of the solar system, returning to the initial state or going back to the main scene of the geography lesson. Fig. 11 shows the Solar System.

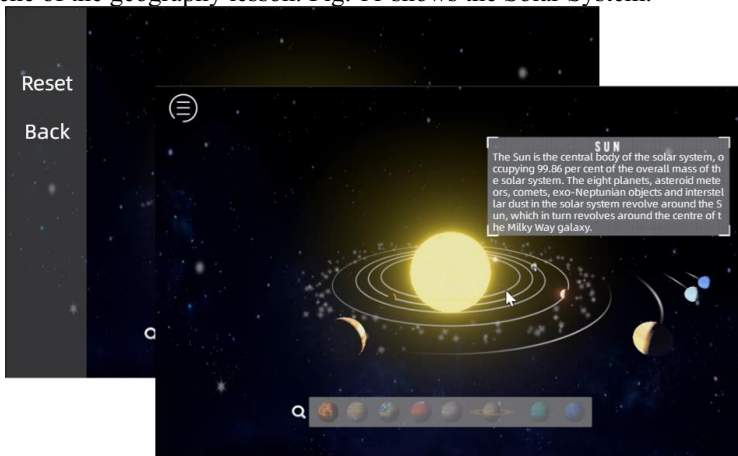


Fig. 11. "Solar System" scenario, click on the planets, the UI interface shows detailed knowledge, click on the upper left corner to return or reset

Clicking on the "Earth Structure" option reveals a three-dimensional cross-section of the Earth in the scene. Users can use mouse clicks and drag to carefully observe and learn about the internal structure of the Earth. Clicking on different parts of the Earth – the core, outer core, mantle, and crust – will display detailed knowledge points on the interface. Clicking on the teacher will provide synchronized voice explanations. Additionally, the presentation screen simulates displaying slides and videos. Clicking on the mouse can enlarge the display of slides and videos on the UI interface. After clicking on the slides, the arrows on both sides of the interface can be used to switch between different pages of the slides. In the lower-left corner of the interface, there are three icons for controlling video playback, from left to right: start, pause, stop. Moreover, the system includes voice recognition functionality, allowing users to control the display and disappearance of elements through voice commands such as "open/play slides" and "open/play/pause/stop video."

Biology Class Interface: As shown in Fig. 12, this is the main interface of the biology lesson. The UI interface of this teaching scene is essentially the same as that of the geography lesson. In the upper-right corner, there is a dropdown list with three options: "DNA Structure," "Heart Structure," and "Cell Structure." Choosing different options will display different teaching elements.

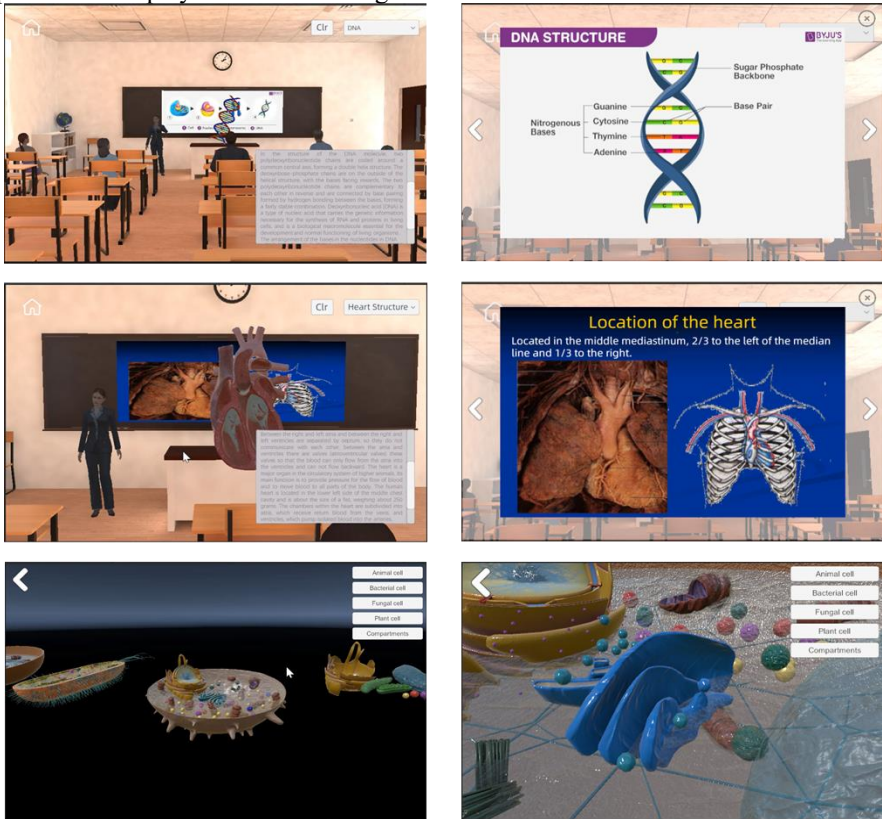


Fig. 12. Biology class scene

Chemistry Class Interface: As shown in Fig. 13, this is the main interface of the chemistry lesson. The UI interface of this scene is slightly different from the previous two courses. When clicking on "Chemical Molecular Structure," 3D models of carbon molecules, benzene rings, and water molecules will appear on the teacher's desk. Clicking on the "Potassium Permanganate Produces Oxygen" option will display experiment steps UI buttons at the top of the UI interface. Users can switch to observe different steps of the experiment by clicking the buttons, and there are explanatory texts on the buttons to assist users in understanding.

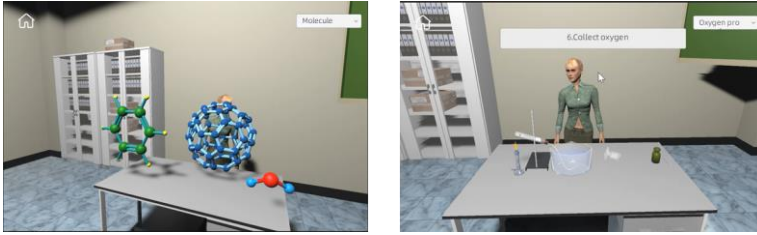


Fig. 13. Chemistry class scene

5.4 DSL Script Control

As shown in Fig. 14, clicking on the button in the bottom-left corner of the UI interface opens the DSL interactive script panel. On the right side of the panel, there are six GUI function buttons: load object, set position, size, rotation angle, set animation, camera, save command, and clear input. Clicking the buttons automatically fills in the code, and users only need to enter values in the corresponding positions to complete the programming, which is advantageous for users with limited programming experience. After entering syntax-compliant instructions, clicking "Save" allows the system to execute the instructions correctly, facilitating users in quickly arranging the Metaverse Smart Classroom.

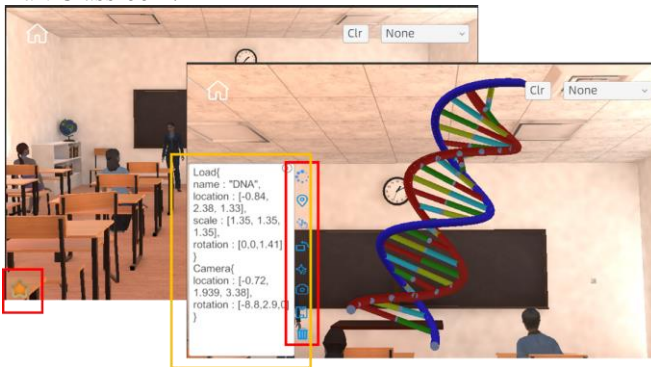


Fig. 14. DSL Script Interaction

5.5 Account Management System

This system, based on the Spring Boot-Vue framework, presents directly to users on the browser side, achieving the decoupling of front-end and back-end for student account management.

As shown in Fig. 15, the system login interface displays a red prompt if the user inputs an account and password that do not meet the requirements. After entering the correct account and password, clicking the login button takes the user to the system's

homepage, which defaults to the personal center, as shown in Fig. 16. Clicking the dropdown button in the top-right corner of the page allows the user to switch to the user center under "Personal Center" or log out under "Logout," automatically returning to the user login interface for re-login. Fig. 17 displays the interfaces logged in by users of different levels.

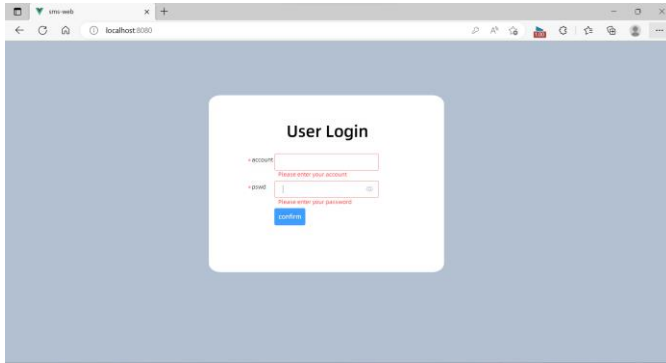


Fig. 15. Login Interface

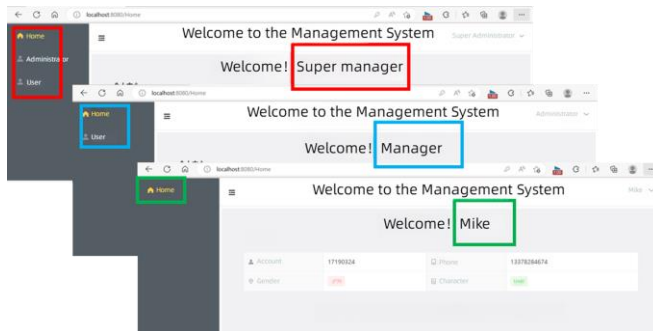


Fig. 16. Personal centre with three different management levels of accounts

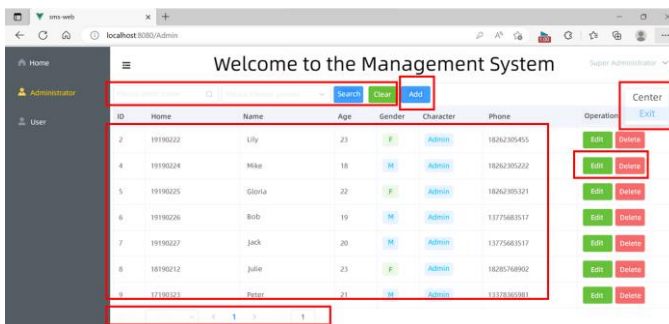


Fig. 17. "Administrator Management" Interface

6 Experiment Results and Analysis

6.1 Educators Classroom Construction Experiment

The DSL interactive script tool designed in this work facilitates the creation of highly personalized intelligent classrooms by non-professional users in a short period. To assess its learnability and effectiveness, a user study was conducted, involving 30 experts from diverse fields who were invited to engage in learning and constructing intelligent classrooms.

Experimental Procedures: This study recruited 30 experts from three distinct domains, namely geography, biology, and chemistry. All participants possessed extensive expertise in their respective fields but lacked programming knowledge.

Firstly, participants were required to acquire foundational knowledge of DSL. We provided an online tutorial covering the syntax, command structure, keywords, and functionalities of DSL. Secondly, practical exercises were offered to assist participants in applying their acquired DSL knowledge. This included tasks such as writing simple DSL scripts and creating virtual environment elements like scenes, 3D models, or interactive elements, actively involving participants in practical virtual environment construction. Three tutors were assigned to provide individual guidance and answer queries, facilitating a better understanding of the learning process.

Subsequently, participants personalized the construction of intelligent classrooms based on their domain-specific expertise and the DSL knowledge acquired during training. The entire process was meticulously documented.

Lastly, participants were required to complete a 7-point Likert scale based on NASA-TLX to assess the usability and efficiency of the system. Following the conclusion of the experiment, semi-structured interviews were conducted to gather participants' feedback and engage in in-depth discussions about system functionalities, points of confusion, and instances of failure. The entire experimental process was recorded, with a total duration of approximately 1.5 hours. The results of educators feedback are shown in Fig. 18.

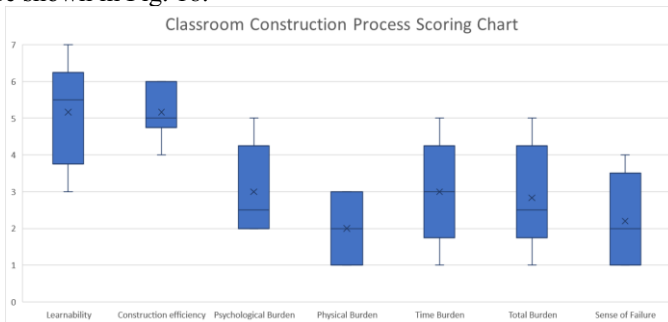


Fig. 18. Educators ratings of the intelligent classroom construction

Experimental Results:

System Usability. Educators generally express a positive attitude towards the usability of the system. They believe that the system offers an intuitive interface and tools, enabling them to relatively easily construct virtual environments. The use of DSL eliminates the need for an in-depth understanding of programming, significantly streamlining and expediting the classroom construction process.

Functional Evaluation: Educators have provided positive evaluations regarding the functionality of the system. The system offers a diverse set of tools for virtual environment design, allowing them to create customized classrooms and interactive elements that align with their teaching needs. The flexibility of DSL is acknowledged for providing a broad range of customization options, enabling effective communication of instructional concepts.

Satisfaction: Educators, overall, express satisfaction with the system. They believe the system provides innovative and engaging teaching tools capable of sparking student interest and enhancing the learning experience. The flexibility and interactivity inherent in the system contribute to a more appealing and effective teaching process.

6.2 Experimental Comparison of Student Interactive Behavior and Learning Performance

To assess the impact of using DSL to create virtual classrooms versus traditional educational methods on student interaction behavior and learning performance, we engaged 30 students in a user study.

Experimental Procedures: *We established two different classrooms in the virtual environment: one constructed using DSL (Domain-Specific Language) and the other representing a traditional classroom setting. Students were randomly and equally assigned to both groups. The experimental group engaged in self-directed exploratory learning within the DSL-created virtual classroom, while the control group received the same educational content in a traditional classroom environment. Student interaction behaviors, including dialogue content, posed questions, and completion of learning tasks, were recorded through game logs in the virtual environment. Additionally, online quizzes were administered to gather data on students' learning performance.*

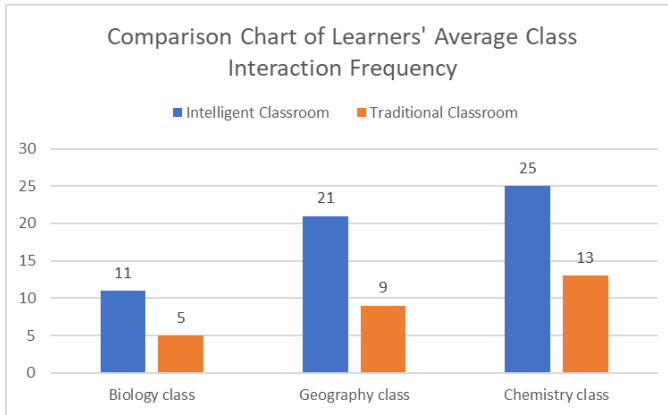


Fig. 19. Comparison of learners' average classroom interaction

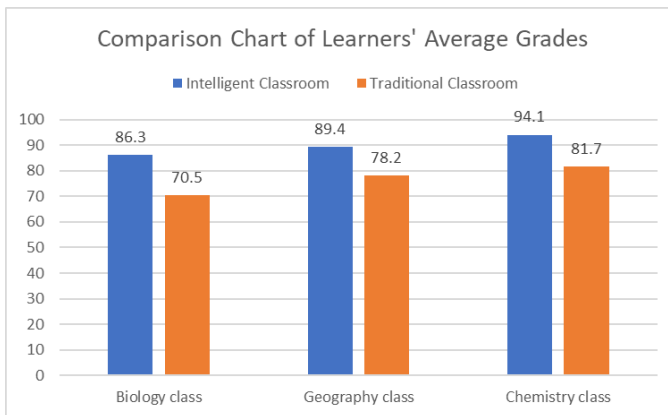


Fig. 20. Comparison of learners' average academic performance

Experimental Results: *As can be seen from Fig. 19, students in the Experimental Group exhibited increased interaction within the Virtual Classroom, including more frequent questioning and active participation in discussions. Additionally, Fig. 20 shows that the Experimental Group displayed an improvement in learning performance.*

6.3 Analysis and Discussion

The Metaverse Intelligent Classroom based on Domain-Specific Language (DSL) has achieved significant success. Learners and educators alike express that the design of DSL makes it easier for them to create and customize Metaverse learning environments. The personalized learning experience enhances learner engagement and motivation, contributing to improved learning outcomes. Furthermore, the interactivity of

DSL makes learning more enjoyable and increases students' depth of understanding of knowledge.

7 Conclusions

This paper proposes a method for constructing a Metaverse Smart Classroom based on DSL. Through a detailed exploration of system architecture, system design, implementation results, and more, the paper unveils the design principles and practical applications of this approach. The DSL interactive scripting tool we designed proves beneficial for users lacking programming experience, enabling them to efficiently customize a personalized virtual educational environment based on their specific needs. This approach aims to provide a more immersive, flexible, and personalized learning experience for both students and educators. The potential impact of this method includes advancing virtual education, enhancing educational quality, and opening up new opportunities and possibilities in the field of education.

References

1. Zheng Z, **e S, Dai H N, et al. Blockchain challenges and opportunities: A survey[J]. International journal of web and grid services, 2018, 14(4): 352-375.
2. Sagioglu S, Sinanc D. Big data: A review[C]//2013 international conference on collaboration technologies and systems (CTS). IEEE, 2013: 42-47.
3. Negnevitsky M. Artificial intelligence: a guide to intelligent systems[M]. Pearson education, 2005.
4. Malik M I, Wani S H, Rashid A. CLOUD COMPUTING-TECHNOLOGIES[J]. International Journal of Advanced Research in Computer Science, 2018, 9(2).
5. Dincelli E, Yayla A. Immersive virtual reality in the age of the Metaverse: A hybrid-narrative review based on the technology affordance perspective[J]. The Journal of Strategic Information Systems, 2022, 31(2): 101717.
6. Motah M. Learning, types of learning, traditional learning styles and the impact of e-learning on the performance of secondary school students: the perception of teachers[C]//Proceedings of the 2007 Computer Science and IT Education Conference. 2007: 483-498.
7. Shu X, Gu X. An empirical study of A smart education model enabled by the edumetaverse to enhance better learning outcomes for students[J]. Systems, 2023, 11(2): 75.
8. Biocca F, Delaney B. Immersive virtual reality technology[J]. Communication in the age of virtual reality, 1995, 15(32): 10-5555.
9. Lele A. Virtual reality and its military utility[J]. Journal of Ambient Intelligence and Humanized Computing, 2013, 4: 17-26.
10. Bown J, White E, Boopalan A. Looking for the ultimate display: A brief history of virtual reality[M]//Boundaries of self and reality online. Academic Press, 2017: 239-259.
11. Croft F M. The history of the international conference on geometry and graphics—one person's reflection[J]. J. Geom. Graph, 2016, 20(2): 253-262.
12. Brooks F P. What's real about virtual reality?[J]. IEEE Computer graphics and applications, 1999, 19(6): 16-27.
13. Lieberman H. Intelligent graphics[J]. Communications of the ACM, 1996, 39(8): 38-48.

14. Lenoir T. All but war is simulation: The military-entertainment complex[J]. *Configurations*, 2000, 8(3): 289-335.
15. Blaine A M. Interaction and presence in the virtual classroom: An analysis of the perceptions of students and teachers in online and blended Advanced Placement courses[J]. *Computers & Education*, 2019, 132: 31-43.
16. Abramczuk K, Bohdanowicz Z, Muczyński B, et al. Meet me in VR! Can VR space help remote teams connect: A seven-week study with horizon workrooms[J]. *International Journal of Human-Computer Studies*, 2023, 179: 103104.

Open Access This chapter is licensed under the terms of the Creative Commons Attribution-NonCommercial 4.0 International License (<http://creativecommons.org/licenses/by-nc/4.0/>), which permits any noncommercial use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.

