Virtual Mechanical Assembly Training Based on a 3D Game Engine

EL-Sayed S. Aziz[1], Yizhe Chang[2], Sven K. Esche[3] and Constantin Chassapis[4]

[1]Stevens Institute of Technology and Mansoura University, eaziz@stevens.edu
[2]Stevens Institute of Technology, ychang1@stevens.edu
[3]Stevens Institute of Technology, sesche@stevens.edu
[4]Stevens Institute of Technology, cchassap@stevens.edu

ABSTRACT

Designed for personal computers, 3D video games are powerful tools with respect to graphics rendering, real world physics simulation, human-computer interaction and multi-user communication. Because of these favorable features of 3D games, their adaption for serious applications has been widely researched recently. These applications generally focus on topics such as real world scenario reconstruction, which require no or only minor development efforts on the game engines themselves. Contrary to this, the development of virtual education and training environments requires the integration of complex engineering systems into games, which poses greater challenges and thus causes this topic to be covered less frequently. This article presents a framework for authoring virtual environments (VEs) for mechanical assembly training using a commercially available 3D game engine. The VE presented here allows multiple users to conduct simulations of assembly procedures in a collaborative manner and provides an immersive user experience with user-friendly human-computer interactions. In order to enable diverse types of assemblies, the VE uses feature-based representations of assemblies. CAD concepts such as form features, parts, kinematic joints and sub-assemblies are modified for implementation into the VE. This framework is explained through a sample assembly process of a planet gear train. The results of a study conducted in an undergraduate mechanical engineering laboratory are summarized briefly. From the study results, it can be concluded that this VE has the potential to become a valuable education and training tool for users, helping them to acquire mechanical assembly skills that can be applied to fields such as manufacturing, maintenance and repair.

Keywords: virtual assembly, virtual environments, assembly training, assembly constraints, 3D game engine, gear train.

1. INTRODUCTION

In recent years, the level of realism in virtual environments (VEs) has increased dramatically, from 2D drawings to highly detailed 3D visualizations. These and other capabilities have led to virtual technologies becoming recommended practice and preferred tools for solving problems across a number of businesses. Training based on VEs is especially suitable for simulating procedures involving dangerous tasks [16]. Furthermore, VEs provide a low-cost alternative to training using real physical systems. In addition, VEs have also been developed as a potential solution for teaching practical and technical skills in environments that are safe, and they allow for users to make and learn from mistakes without consequences. As the technology of VEs progressed, it was adopted for use in many fields, including education, manufacturing operations, maintenance of complex equipment, medicine, architecture, city planning, disaster response and military training. Recently, VEs have been widely used to create digital prototypes in order to save time and money spent on product development and manufacturing of physical artifacts. The most distinct characteristics of VEs are that they are designed to provide users with a sense of immersion and to facilitate interactions between users and objects [44].

Multi-user computer games have gone from a novelty item to being a part of everyday life, serving for instance as learning devices for people of all ages. As a result of the widespread availability and popularity of gaming technologies, researchers from

Taylor & Francis
Taylor & Francis Group

different areas have been designing game-based VEs for improving teaching or training practice. Such systems must combine instructional strategies and ludic activities to achieve educational goals [2]. Some of these systems reconstruct certain real scenarios while others simulate real environments. As examples for the former type, there are reports of using video games for fire drill [31], driver [12] and pilot [24] training, medical training, [19] and military skills and tactics training [25]. Examples for the latter type include virtual museums [40], virtual architectural tours [34,41] and virtual classrooms [37]. Recently, classroom multi-player presential games have been introduced to replace slide shows by bringing virtual worlds directly into classrooms. Starting in 2007, game-based interactive engineering laboratories were designed and implemented in a junior-level engineering course [6]. In the recent past, some scholars have started to develop a virtual assembly system based on the Unity3D game engine [29] and others have utilized the Torque engine to develop a virtual design review system [15].

For assembly training VEs such as the one presented here, other factors such as the involvement of multiple participants who play different roles as well as the mechanical modeling and the interactions between the participants and mechanical components also need to be considered [8]. This paper introduces different methodologies for adopting computer game technologies such as the "Source" game engine (used in "Half-Life") to develop VEs for mechanical assembly training. A successful implementation of a virtual mechanical assembly environment may draw on the experiences from both the contemporary video game industry and modern computer-aided design/manufacturing research.

2. LITERATURE REVIEW

Virtual reality (VR) is a computer simulation with its key elements being virtual world, immersion, sensory feedback and interactivity. Therefore, VE can be perceived as a part of VR with an emphasis on the virtual world and interactivity. In this section, a review of VR related literature by category is provided.

2.1. Desktop Computer (non-immersive) VR Systems

Desktop computer (non-immersive) VR systems**,** [5] in which the user sees the 3D world on a computer screen and navigates through the space with a control device such as a mouse. Sometimes, the user also wears 3D glasses in order to enhance the visual depth and to create stereoscopic effects. Until now, many desktop VR training systems have been developed because they are inexpensive and easy to use. For instance, a desktop VR system for maintenance training known as V-REALISM was developed [20]. It provides maintenance engineers with a desktop VE for practicing different disassembly operations of the systems under maintenance. As another example, a desktop-based virtual assembly application called Virtual Environment for Design for Assembly (VEDA) that uses physics-based modeling of dynamic interactions among virtual objects was presented [13]. Desktop VR systems are also used as the 3D architectural design planning tool for many Hollywood movies and advertisements as well as for educational applications [14].

Another advantage of desktop VR systems is the Virtual Reality Modeling Language (VRML), which can be used to build virtual 3D scenes and has been adopted as a standard for the transfer of 3D model data and virtual worlds via the Internet. Using VRML and Java, an architecture for virtual laboratory experiments was developed based on which a variety of virtual experiment simulations were implemented [7], including a virtual fluid mechanics laboratory [17] as well as simulations of a vibration system, a beam deflection system and an industrial emulator system [10]. This virtual laboratory provides a full-featured VR user interface that allows the experimental configuration and parameters to be controlled interactively by the students. Furthermore, learning outcomes were assessed by testing the students' knowledge of specific concepts taught in each laboratory [23].

2.2. Immersive VR Systems

Immersive VR systems**,** which often use head-mounted displays (HMD) or projection-based display systems, fiber-optic-wired gloves or wands, modeling software (geometric modeling and physics-based modeling) as well as position-tracking devices or other devices to immerse the users in the virtual world. A well-known application of the immersive VR in assembly design and training environments is the Virtual Assembly Design Environment (VADE) [18]. The VADE system has the ability to extract assembly information from CAD models and captures the design intent. For example, when the user selects a component for assembly with another component, the constraints between the various planes and axes of the components to be assembled are loaded. By relating these planes and axes, the components become constrained, thus preventing certain relative motions. In addition to the constrained motion, a combination of collision detection and gravity effects are used to realistically simulate the behavior of industrial assembly operations. The Virtual Environment for General Assembly Simulation (VEGAS) [36] system uses data gloves from 5DT [46] for implementing human grabbing motions, VR Juggler software [39] for virtual environment management and Voxmap PointShell software (VPS) [38] from the Boeing Corporation for collision detection. The VEGAS system represents a fully immersive Cave Automated Virtual

Environment (CAVE^{TM}) for both complex assembly models and assembly operations, which provides engineers with an immersive assembly planning and training environment.

Augmented Reality (AR) [30] is a form of an enhanced immersive VR system, which can mix real and virtual features to improve the users' feeling of presence. This is achieved by placing virtual objects or information cues into the real world as perceived by the user. One important application of AR is spatial information systems, which are used for example for exploring urban and planetary environments in space. Another important application of AR is in industrial manufacturing. Boeing researchers found that AR technology can provide an effective method for simulating many possible scenarios and multiple configurations without the need to fabricate physical artifacts [22]. In the field of education, AR was used to enhance the teaching of engineering graphics by enabling the students to visualize and interact with their designs from different points of view in a more intuitive way [35]. Then, the students can plan the assembly process concurrently and check for fitting and interference between components [28].

2.3. Semi-Immersive VR Systems

Semi-immersive VR systems represent a relatively new development in the world of VR technology. They provide a greater sense of presence and a greater appreciation of scale than non-immersive systems. Such systems are comprised of a relatively high performance graphics computing system, which can be coupled with a large monitor or a large-screen projector system in order to increase the feeling of immersion or presence experienced by the user. They provide the user with an experience similar to that of an IMAX theatre. The viewers do not need to wear virtual reality kits such as a data gloves or HMDs, and thus they are still aware of the real world outside of the VE. Sample applications of semi-immersive VR include virtual assembly systems and flight simulations [1]. While semi-immersive VR systems have a few advantages over fully immersive systems (they are less costly and easier to use), they also exhibit some disadvantages, namely a limited range of interaction devices and problems with multi-user applications.

However, even though in most instances VR is designed to simulate the real world in a realistic fashion, it still has some shortcomings to be overcome. First, while VR eliminates the physical distances between the users and/or objects in the VE, the users' real identities are lost, thus decreasing the credibility of the users and the VR itself. Another shortcoming is that the users cannot move around in the VE in a natural way as they would in the real world. They are either constrained to a fixed position or can only move in a limited space. Furthermore, VR is capable of coordinating the users' interactions, both spatially and temporally, but fails to induce the feeling of presence. Note that the essence of the above shortcomings of VR follows from the word 'virtual'.

Virtual training environments implemented using advanced game technologies have the potential for gaining wide acceptance by diverse industries. These technologies lead to a significant reduction in computer hardware cost, increase the social acceptance of applying game-based tools to serious applications and facilitate the implementation of innovations in the adopting industries. Modern game engines enable verbal communication by supporting headsets and microphones, the control of the movement and orientation of avatars, the modification of the avatars' appearance according to the users' preferences, the design of the environment consisting of simulated objects that can be grabbed or moved as well as data input by keyboard or mouse [21]. Moreover, game engines are also equipped with artificial intelligence to guide computer-controlled player avatars in the game and to enhance the social dynamics between the players. This provides a new prototype for human-computer interaction by immersing the players into the game-generated scene and allowing them to interact with computers. Advanced gaming technologies address the value of social rules, norms and expectations that are crucial to interpersonal relationships even when players interact via computers. For example, avatar-based advertising in Second Life offers a promising corporate communication channel for brand marketing and interactive advertising [4]. Spokes-avatars or sales-avatars are used as company representatives, personal shopping assistants, conversation partners and recommendation agents who interact with participants in a virtual retail store.

3. GAME ENGINES FOR VE AUTHROING

3.1. Game Engine Selection

Game engines provide a new option for human-computer interaction by immersing the game players into the game-generated scenes and allowing them to interact with the game objects and with other players in a natural fashion. This capability makes game engines a perfect tool for creating virtual assembly training environments that involve interactions between multiple players, something that proves difficult to achieve when using traditional computer simulation techniques. Therefore, computer games can support collaborative learning activities and provide the players with the feeling of social presence with teammates as well as physical presence in the facility.

Multiple elements are essential for 3D games. These elements include elaborate backgrounds, appropriate challenges, immersive graphical/audio effects, realistic physical simulations as well as strong engagement of players [33]. From a technical perspective, the infrastructure of 3D computer games can be classified into a system layer, a game engine layer and a game play layer [3]. The system layer mainly consists of the hardware and the operating system that support the operation of a computer. The game engine layer lays out the foundation of game elements such as rendering, physics, audio, artificial intelligence, online communication, etc. This layer provides the game developer ready-to-use functions and utilities for implementing complex gaming effects. The game play layer, which is at the top of these three layers, includes features such as avatars, levels, etc. The development of this layer requires combined efforts of computer technicians and, more importantly, artists.

Although game engines were developed for entertainment purposes, their capabilities can be utilized to support the design of educational applications. These advantages include:

- Software Development Kits (SDKs): Many game engines provide free SDKs and associated documentation to facilitate game design. Sometimes they also include tutorials [32].
- Scripting languages: While some game engines require the usage of specifically designed computer languages for game development, most of them allow developers to use popular languages such as C + +, Lua, Python, etc. for game editing. In addition, most game engines allow third-party add-ons in the form of dynamic linked libraries.
- Low communication latency: In order to fulfill the needs of online gaming, most game engines are optimized for transmitting data between the client and server computers efficiently.
- Immersive and realistic environments: In order to attract players, multiple state-of-the-art rendering techniques are applied to provide accurate visual details as well as to reduce the required rendering time. These technologies include culling, per-pixel shading, etc. Game engine developers also make efforts to improve physical effects by considering real-world kinetics, high resolution collision detection, etc.
- User-friendly interface: Games are typically designed such that its users do not require special training to play them.

Game engines exhibit some limitations that make their application for the implementation of assembly procedure simulations challenging. The existing physics engines usually significantly over-simplify the actual physics. For instance, while the effect of gravity is modeled, the gravitational acceleration is usually ignored. Similarly, the collision models are usually restricted to a convex envelop, thus neglecting any existing concave portions of the objects. However, given the limited level of precision required in physics simulations for education and training, game engines are still viable development tools.

3.2. Using GMod for Authoring an Assembly VE

"Garry's Mod" (GMod) was selected for authoring an assembly VE. GMod is a physics sandbox of the "Source" game engine, which powers widely played shooting games such as "Half-Life" and "Counter-Strike". Originally designed to enable game players to try their own "contraptions", GMod supports Lua scripting, which allows VE developers to create things beyond that. Lua scripting can be applied for various purposes such as setting the mass of models (called scripted entities in games), programming their kinematic behavior, triggering events after certain conditions are met, etc. By programming in Lua, the CAD features of an assembly can be incorporated into the game engine [11].

However, Lua is less powerful in geometry modeling. GMod uses polygonal meshes to represent models. Therefore, models of mechanical parts usually consist of hundreds or even thousands of vertices, which renders them too complex for scripts to manipulate. Contrary to Lua scripts, professional 3D CAD software (e.g. SolidWorks) is suitable for creating geometric models. After adding textures and bounding volumes to these models, they can then be exported to the game. The "Source" game engine also provides a map editor, called "Hammer", for game developers to edit the surrounding environment (e.g. walls, doors, windows and lights).

3.3. Game Engine Physics and Collision Detection

The "Source" game engine contains an embedded physics engine for physics simulation and collision detection. For physics simulations, the game engine automatically computes the entities' physical state variables (e.g. velocity, acceleration, etc.) based on their physical properties (e.g. mass, moment of inertia, etc.) and the applied loads. Software developers are also provided with Lua scripts for manipulating the entities' parameters.

In the game engine, collision detection is achieved by the convex polyhedral method. Each entity, along with its geometry mesh, also has a physics mesh, which can consist of one or several convex polyhedrons. At each physics simulation cycle, the game engine executes a polyhedron collision detection algorithm to check whether two entities have collided or not. In order to accelerate the collision detection process, a game map is usually partitioned into several blocks, and collision detection only happens between entities in the same block. If an entity's shape is close to a rectangular cuboid (e.g. a wall or a door), a rectangular cuboid physics mesh with 6 rectangular faces, which is a special form of a polyhedron, is used to further simplify the detection. This type of cuboid is called a "bounding box" in the game developer community.

4. REPRESENTATION, GENERATION AND ANALYSIS OF MECHANICAL ASSEMBLY SEQUENCES

4.1. Elements of Mechanical Assemblies

An assembly must at least contain the following elements [26]:

- *Mechanical Parts* are solid 3D geometries that cannot be further split (e.g. gears, shafts or bearings).
- *Assembly Configurations* represent the hierarchical relationships between the components (assemblies, sub-assemblies and parts).
- *Part Associations* provide detailed information for the joints between pairs of connected parts. In feature-based assembly, part associations are implemented on the basis of feature association.

A planetary gear train system is used here as an example to illustrate the proposed method. Although planetary gear trains are fairly simple assemblies, they cover the main concepts introduced in this article. Fig. 1 shows an exploded view of the main parts of a planetary gear train assembly.

4.2. Assembly in CAD Systems

In the research described here, the virtual assembly model of a planetary gear train system was established based on the real assembly sequences of the planetary gear system. The data captured in the assembly of the planetary gear train model includes:

- *Information about planetary gear train components*, which are the 3D components such as gears, shafts, base, etc. that are used here in developing the assembly tree to represent the sequence of a planetary gear system assembly as required in the real application.
- *Information about standard components,* such as nuts and bolts, keys, electric motors, etc. For instance, four screws are used here to fasten the base to the table.
- *System configuration*, which represents the hierarchical relationships between the components (assemblies, sub-assemblies and components).
- *Feature associations*, which provide detailed information on associations between pairs of physically connected components, between pairs of components that are not physically connected and detailed information about assembly features (see Fig. 2). In the figure, components (assemblies, sub-assemblies and components) are shown by blue boxes with gray background, mating assembly features used for defining the connectivity relationships between components of the planetary gear train system are described by purple hexagonal shapes, mating constraints and joint attributes between mating components are shown by orange circles and kinematic constraints of kinematic pairs (revolute, prismatic, screw, planar, sliding surface, rolling surface, etc.) are defined by blue rectangles.

Depending on the type of the association [27], the pairs of physically connected components can be classified as follows:

- *Fixed connections (FC)* such as rigid joints. The pair is physically connected and fixed. Here, for instance, the planet gear pin fits into the hole of the planet carrier with a tight fit (see Fig. 2). The chamfered hole should be positioned such that it is facing upward in order to allow for easy alignment and insertion of the pin into the hole.

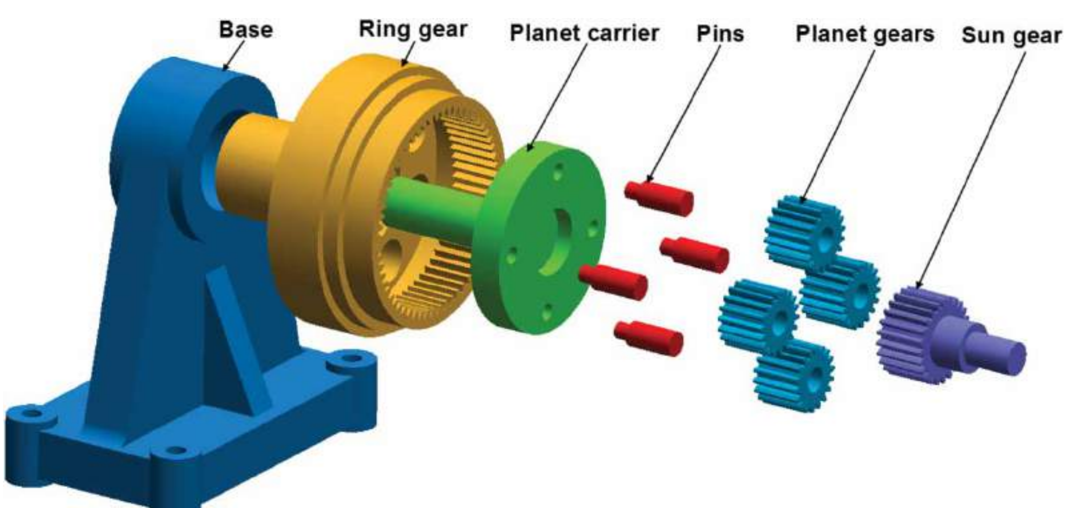

Fig. 1: Exploded view of planetary gear train assembly [8].

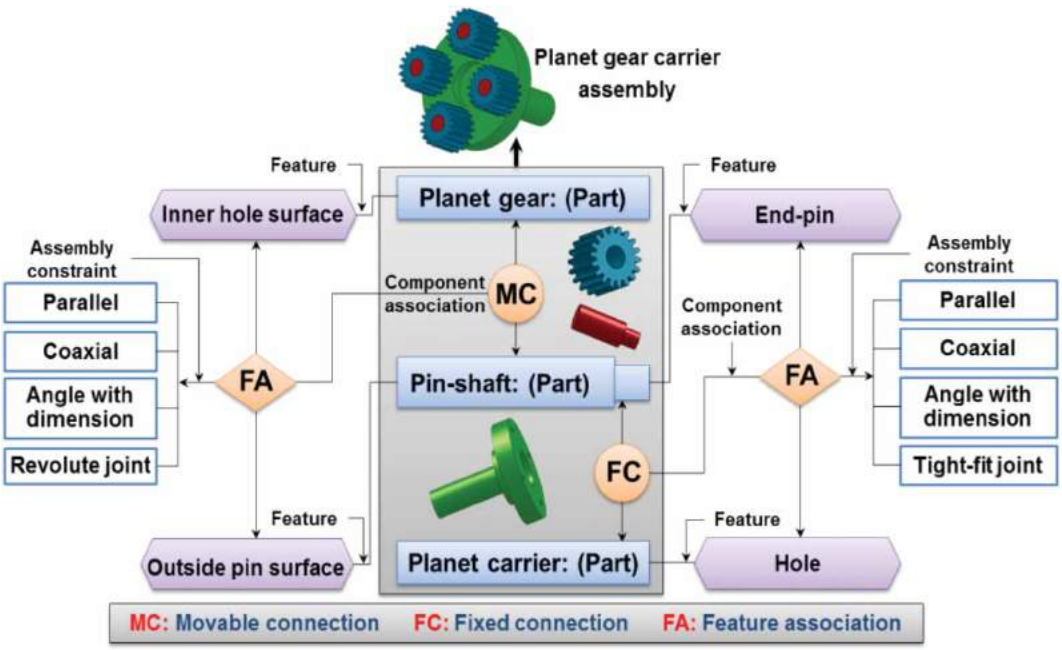

Fig. 2: Assembly relationships among planet gear carrier components [8].

- *Movable connections (MC)*, which describe associations when the pair is physically connected but the components are movable relative to each other. Here, the four planet gears are assembled by a loose fit (revolute joint) with their respective planet gear pins, which in turn are inserted into the planet carrier (see Fig. 2).

Pairs of components that are not physically connected can be classified as follows:

(a) *Relative motion* between a pair of components. Here, the four planet gears of the planet gear carrier assembly are simultaneously meshed with the sun gear and the outer ring gear (see Fig. 2). Consequently, these planet gears are geometrically constrained to roll on two surfaces at once. The distance that the planet gears roll on the inside of the ring gear equals the product of the rotation angle of the planet carrier and the pitch radius of the ring gear. Therefore, there are several configurations that can produce a wide range of speed ratios depending on the chosen driving gear and the part constraints.

(b) *Relative position and orientation* of a component to another component. Here, the function of the planet carrier is to maintain a fixed distance between the planet gears that are equally spaced around the perimeter of the sun gear in order to distribute the transmitted load (see Fig. 2). Four identical planet gears are configured to rotate together with the carrier about the axis of rotation. Each planet gear includes a pivot axis that is concentric to a hole in the carrier, thus allowing for rotation of each planet gear about its pivot axis relative to the carrier.

- *Assembly features* [45] are particular form features, which represent a region of a component that is of interest in the assembly context. They describe the details of the components' associations such as the component locations and orientations, mating faces, joint characteristics, relative degrees of freedom, insertion trajectories, cross-sectional properties and assembly sequencing information. Fig. 3 depicts the assembly relationships between the ring gear base assembly and the planet gear carrier assembly.

Fig. 3 depicts a specific link for each valid component-component-connector combination. Specifically, the output shaft diameter of the planet carrier must equal the hole diameter of the ring gear. The output shaft's length should be greater than the hole depth. A clearance fit is required between the output shaft of the planet carrier and the inner hole of the ring gear in order to permit relative motion. In the game environment, an assembly validation system checks the link information and component information against the assembly conditions from the assembly database in order to verify the feasibility of the assembly. Once the end shaft of the planet carrier enters the hole, the outside surface of the shaft and the inner surface of the hole are aligned in order to facilitate the assembly of the entire planet gear carrier into the ring gear. At the same time, the teeth of

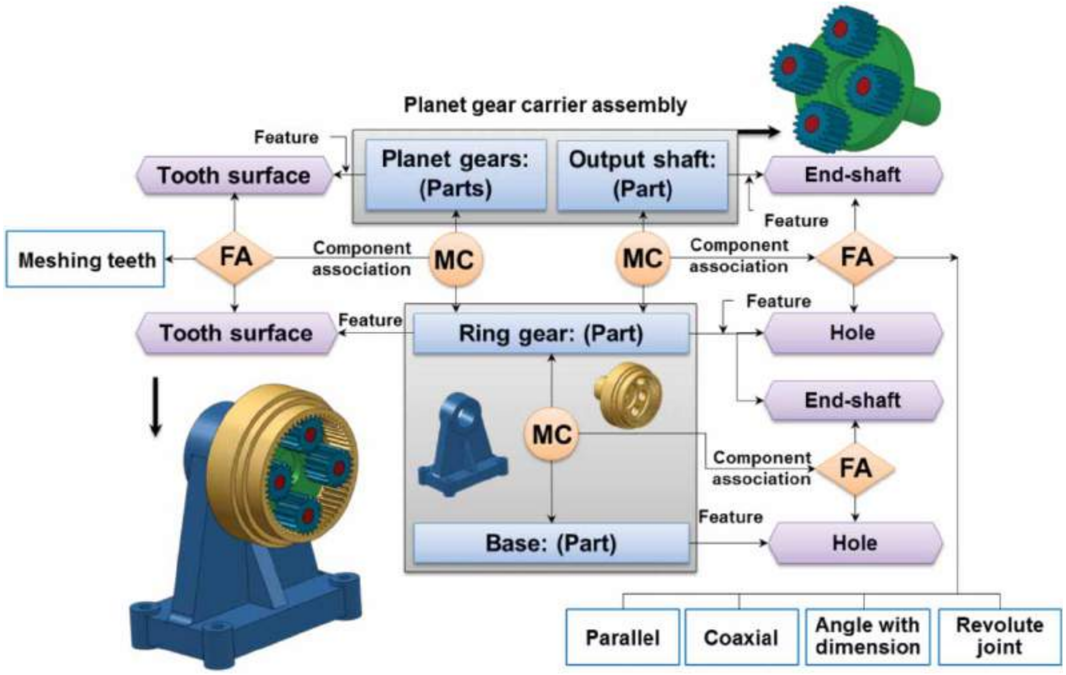

Fig. 3: Assembly relationships between ring gear base assembly and planet gear carrier assembly [8].

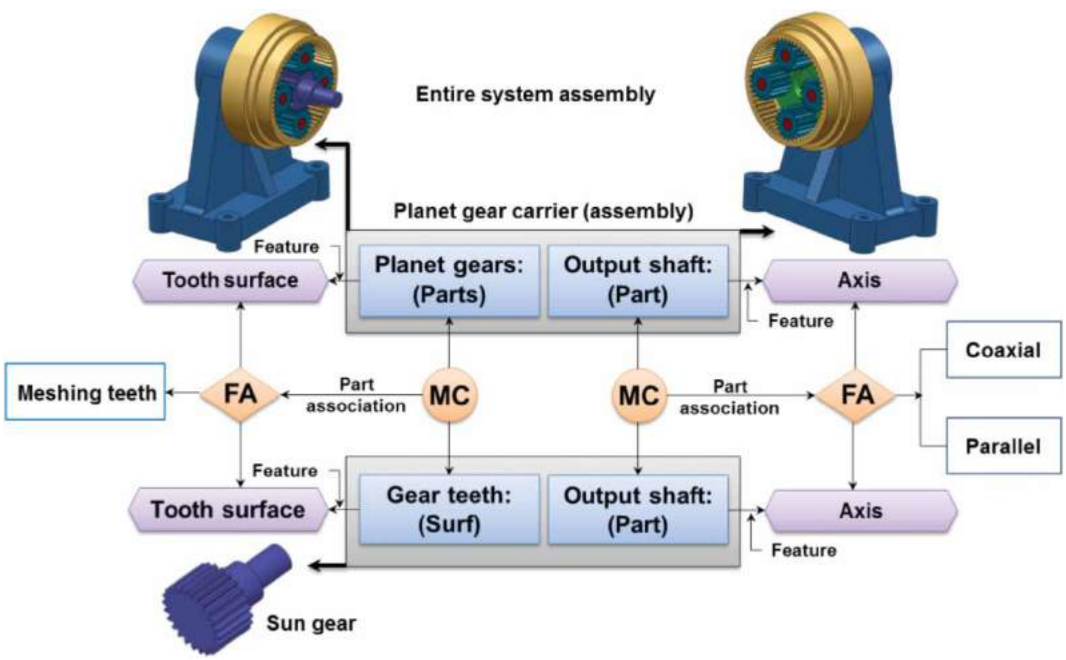

Fig. 4: Assembly relationships between sun gear and planet gear carrier assembly [8].

the planet gears are aligned with the internal tooth spaces of the outer ring gear in order to mesh the planet gears with the outer ring gear. In the final step, the sun gear is placed in the center of the assembly and its teeth are aligned with the teeth of the four planet gears in order to facilitate the meshing and assembly. The final structure of the assembly of the planetary gear train setup is illustrated in Fig. 4.

5. ASSEMBLY IN GAME ENVIRONMENT

5.1. Features and Feature Slots

Although feature-based modeling is prevalent today, geometry models in GMod are still in polygonal representation. In feature-based modeling, a geometry is formed by a combination of elementary features. By this method, mechanical parts, which usually have regular shapes, can be modeled and modified conveniently. However, geometries encountered in games are usually irregular and therefore template-like feature-based modeling cannot meet the needs of most GMod programmers and players.

For a mechanical assembly in GMod, form features are reintroduced. These features are identified on a polygonal mesh. In the current implementation, this identification is performed manually. However, unlike CAD feature-based design in which all form features must be included, this identification may omit the characteristics that are irrelevant to the assembly. This is because on a mechanical part model, most of the geometry characteristics do not serve any assembly purpose (i.e. no part may be paired with them). For instance, the web on the base part (see Fig. 5) does not have a specific function in the assembly, and therefore it is not necessary to identify it as a feature. On the contrary, the cylindrical hole can match a shaft, and thus it is identified as a feature.

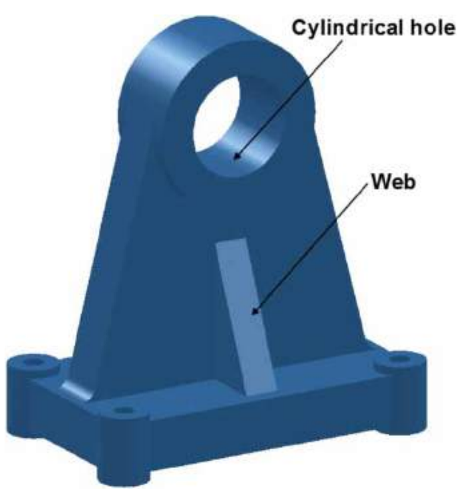

Fig. 5: Feature identification on base part.

A feature slot is a data package (class) that describes a specific feature on a part in detail. Each slot holds attributes of that feature, such as the type of the feature, the position of the feature, etc. Fig. 6 shows the major attributes of a feature slot. The feature_slot class has an association relationship with the feature_type class, which can be further categorized into cylinder feature, cuboid feature, etc. A feature_availability' attribute is devised to prevent one feature slot from establishing feature associations with more than one feature slot simultaneously.

Fig. 7 is an example for the feature identification and feature slot assignment. In a planet gear train, the planet carrier is directly mated with the sun gear, the ring gear and the pins, and thus it has multiple matching feature slots with these parts. In order to mate with the ring gear (which has a cylindrical hole feature), feature slot (1) (which is a cylinder feature) is assigned to the external surface of the cylinder. Furthermore, this feature slot includes the outer diameter d_1 of the cylinder to be mated with the inner diameter of the cylindrical hole in the ring gear. Similarly, slots (3)-(6) are assigned to the cylindrical hole features with diameter d_3 located at their respective local positions on the planet carrier. Analogously, slot (2) is assigned to the interior surface of the cylindrical hole feature of diameter d_2.

5.2. Feature Associations

In CAD, assemblies are implemented using the concept of mates. When two parts are mated, a joint is formed with a certain number of degrees of freedom removed. This concept can be extended to the assembly VE. In order to construct a feature association in the GMod-based VE, firstly, two slots should be specified. Then, the association is established.

Like feature slots, feature associations are data packages. A feature association is defined between two connecting feature slots, whereby the association type is specified as an additional property. The association type is similar to the concept of mates in CAD and is readily provided by GMod (see Fig. 8).

5.3. Hierachy for Assembly Sequence Representation

When assembling two parts, one part is always fixed and the other part is moved into the assembly position. In the real world, in most cases it can be easily decided which part should be fixed and which one should be moved. When a shaft is inserted into a bearing that is fixed to a base, it is “natural” to move the shaft and insert it into the bearing. However, in CAD applications, this detail is usually omitted since CAD users care more about the assembled product rather than the procedure. For instance, moving a heavy base such as to align with a shaft would be acceptable. Contrarily, when a VE is used to simulate the assembly process, this detail must be taken into account because it would be undesirable for some trainees to mistakenly feel that it is feasible to move the base with the bearing in order to mate it with the shaft.

This confusion can be avoided by introducing hierarchical part relationships. A rank is assigned to each part. During assembly, the high-rank part always moves toward the low-rank part. For instance, the base in the example above is fixed on the floor and thus assigned rank 0.

When assigning the ranks to the parts of an assembly, the basic principle is to assign the lowest rank to the foundation part that supports the whole assembly. The highest rank is given to decorative parts or parts that do not support any other parts. Therefore, in most cases the rank is related to the sequence of

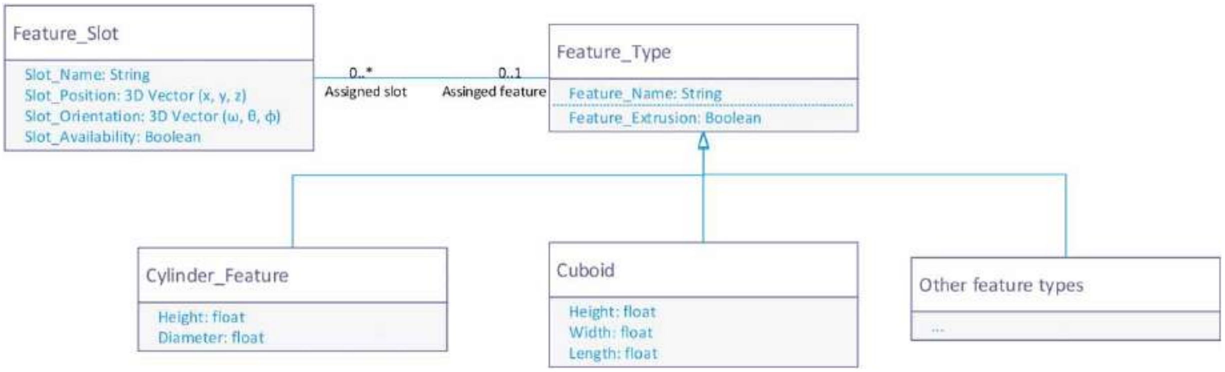

Fig. 6: Major attributes of feature slots and features.

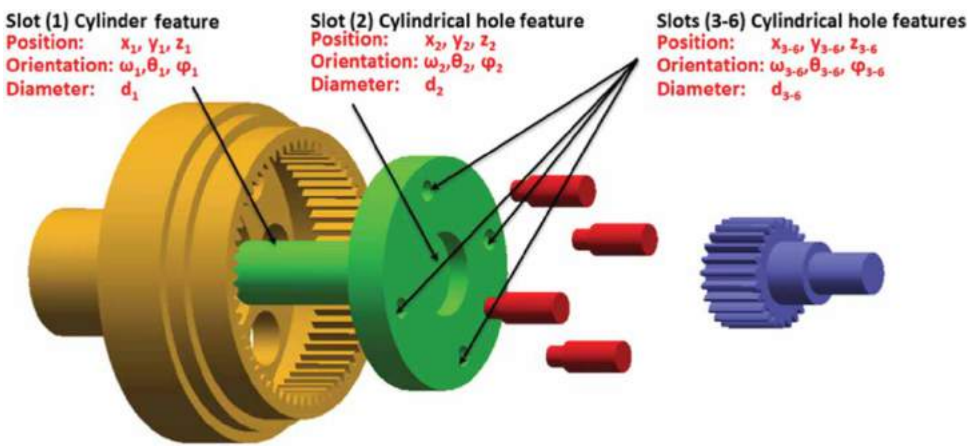

Fig. 7: Features and feature slots of planet carrier.

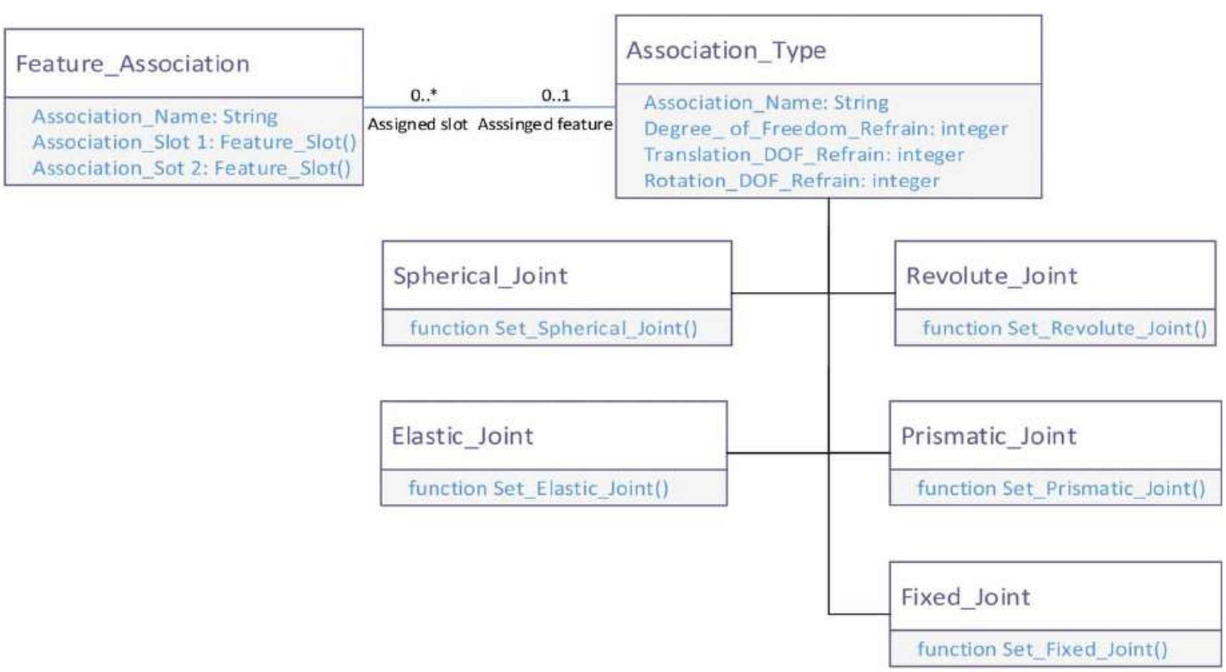

Fig. 8: Feature_association class.

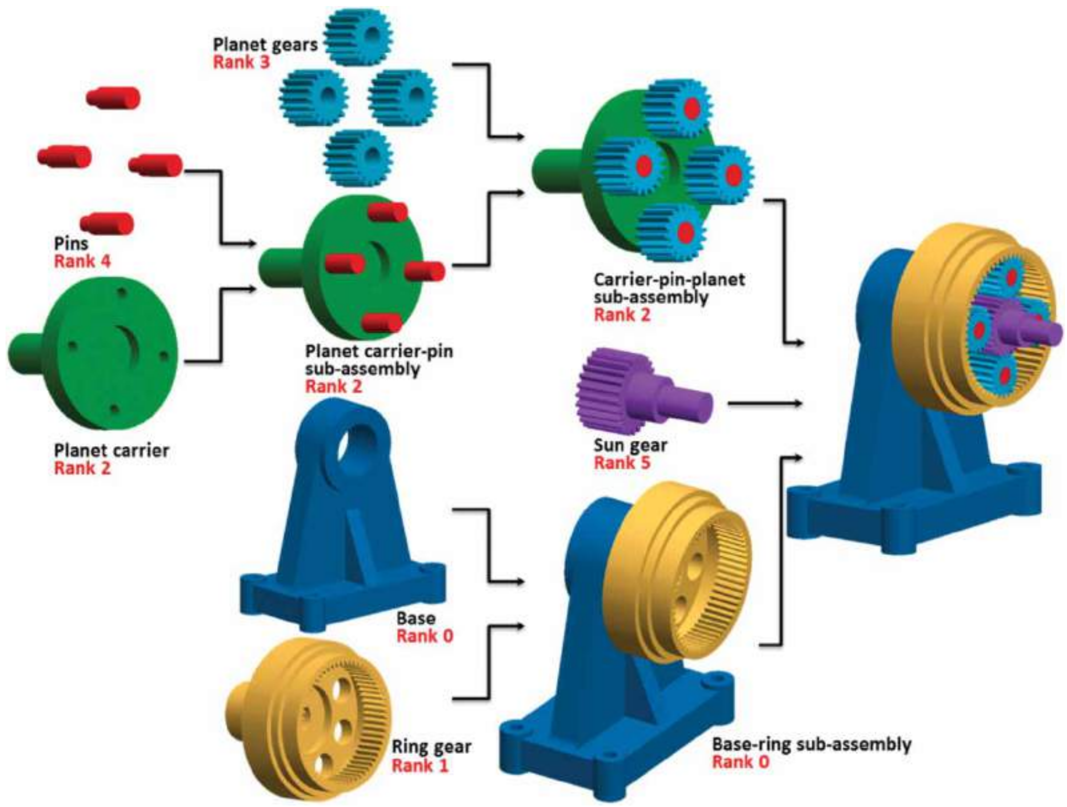

Fig. 9: Hierarchy of gear train assembly.

the assembly. Fig. 9 shows the hierarchy of the planet gear train components. The base part represents the foundation of the whole gear train as all other parts are mounted onto it. Hence, the base has the lowest rank, namely rank 0. The ring gear is directly linked with the base by a revolute joint, and it thus has the second lowest rank, namely rank 1. When the ring gear is ordered to be associated with the base, the virtual system automatically moves the ring gear to the base rather than the other way around. The planet carrier, as an intermediate part, is ranked 2, lower than the ring gear and higher than the other parts, since on one the hand, it is attached to the ring gear, and on the other hand, it holds all planet gears and the sun gear.

The rank assignment is not always directly related to the assembly sequence as it may appear. In some cases, two or more allowable assembly sequences exist. Hence, there is a possibility that a rank that is suitable for one assembly sequence may not be suitable for another. For example, the preferred sequence of assembling the planet gears and pins is to first insert the pins into the carrier and then mount the planet gears onto the pins. Therefore, since the rank of the carrier is 2, the rank of the pins should be 3 and that of the planet gears should be 4. However, it is also acceptable to first connect the planet gears and pins to create planet-pin sub-assemblies and then insert these sub-assemblies into the holes in the planet carrier. In this situation, since then the pins have rank 3 and the planet gears have rank 4, during forming the sub-assembly, the planet gears move to the pins rather than the pins being inserted into the holes of the planet gears. In order to better fit this assembly sequence, the rank of the pins should be 4 and that of the planets should be 3. However, this rank assignment would again not be suitable for the first assembly sequence.

The rank of sub-assemblies can be introduced to further resolve this issue. When two parts form a sub-assembly, the rank of the sub-assembly is assigned to be the rank of its highest-ranked part. When another part or sub-assembly is mounted onto this sub-assembly, the rank of the new sub-assembly is used in the comparison rather than the rank of the part to be connected. In the gear train example, the pins and planet gears are assigned rank 4 and rank 3, respectively. When a pin is first inserted into the carrier, the rank of the carrier-pin sub-assembly takes on the rank of the carrier, which is 2. Since the planet gear has rank 4 and the sub-assembly has rank 2, when mounting the planet gear onto the sub-assembly, the planet gear is the lower-ranked part and moves to the fixed pin. If the other assembly sequence is applied wherein a sub-assembly of pin and planet gear is created first, the pin would move to the planet gear to form the sub-assembly since it is the lower-ranked part. The sub-assembly inherits rank 3 from the planet gear.

Then, in the process of assembling the sub-assembly with the planet carrier, because the rank of the sub-assembly is still lower than that of the planet carrier, the latter would be fixed and the sub-assembly would move to the planet carrier, as it would happen in the real world. In an assembly, two different types of parts that are not associated may share the same rank, although this is not recommended. In the planet gear train, the sun gear can have the same rank as the planet gears without affecting the assembly process. However, it is always better to assign a distinct rank to it. In this case, the sun gear is assigned rank 5, a rank that is different from all others (see Fig. 9).

6. ASSEMBLY OPERATION IN GAME-ENGINE BASED VE

In CAD software, users can create an assembly by selecting by mouse contact mates between flat and/or cylindrical surfaces as well as alignment and orientation constraints. In the VE described here, this practice becomes hard to implement since, first, clicking on a specific surface is hard to do technically in the game-based VE, and second, the game-based VE is designed to provide the trainees the immersive feeling that they are grabbing those parts. Choosing the surfaces and matching them cannot provide such reality, though. Most game engines provide a method for simulating grabbing activities. In the virtual environment based on 'GMod', the trainees can hold or grab something by a tool called 'physics gun', which provides the feeling of grabbing and the handling of which is easy to learn.

In the real world, when one wants to assemble two parts, say, insert a pin into the hole of a planet gear thus establishing a revolute joint between them, one can grab the pin and insert it into the planet gear directly. In the game-based environment discussed here, the same procedure is followed. However, in the real world, one can and must align the pin exactly with the central axis of the hole. Unfortunately, the limited precision of the game engine prevents one from doing so by mouse and keyboard, even if one is experienced with the controls of the game.

Therefore, the operation of assembling two parts was simplified in the VE. Trainees can simply assemble two parts by letting them collide. Then, the VE interprets this as a request for assembly and subsequently either rejects or accepts this request. In the former case, no assembly is formed while in the latter case, a feature association is imposed between the two parts.

This process involves three major steps:

- Identification of all pairs of feature slots that are suitable for being assembled
- Determination of the exact assembly position
- Establishment of the feature association and imposition of kinematic constraints on both parts

Fig. 10 illustrates the detailed processes of establishing a feature association by the VE after a trainee requests the assembly of two parts by letting them collide.

In the first step, three tests are conducted. The first test is to find matching feature types. During this test, all feature slots on both parts are searched and those feature slots that have slots on the other part with matching feature type are kept while the remaining ones are excluded from further consideration. If there are no matching slots, the assembly process is terminated automatically and no feature association is established between these two parts. The second test is to check whether the detailed parameters of these type-matched feature slots are compatible. For instance, if a pair of type-matched feature slots consists of a 5 mm diameter cylindrical hole and a 10 mm diameter cylinder, this pair would fail in this test. The last test is to check the occupancy of these type- and parameter-matched slots. In order to qualify for a feature association, neither of the slots can be in occupied state. Matching slot pairs that have passed all three tests are qualified pairs and are passed to the second step.

In the second step, the exact slots for assembly are determined. There are situations where there is more than one qualified slot pair (for instance, the planet carrier has four identical feature slots for pins). Once there is more than one qualified slot pair, one of the following options can be chosen in order to decide which one is used for establishing the feature association:

- Automatically establish the feature association between the first qualified slot pair. Since all feature slots are numbered and thus sorted, the assembly can always start with the first available feature slot. This is the easiest way for trainees.
- Use a pop-up window to ask the trainee to pick a feature slot. By doing so, the trainee can freely choose which feature slot should be used.

Once a pair has been chosen (either automatically or manually), the association is ready to be established, which represents the last step. First, the positions and orientations of the parts are adjusted in order to move them in the correct assembly position and orientation. For example, the pin moves exactly into the selected pin hole of the carrier with correct position and orientation. As mentioned in Section 3, only the lower-rank part is translated and rotated such as to comply with the position and orientation of the higher-rank part. Finally, the feature association is established.

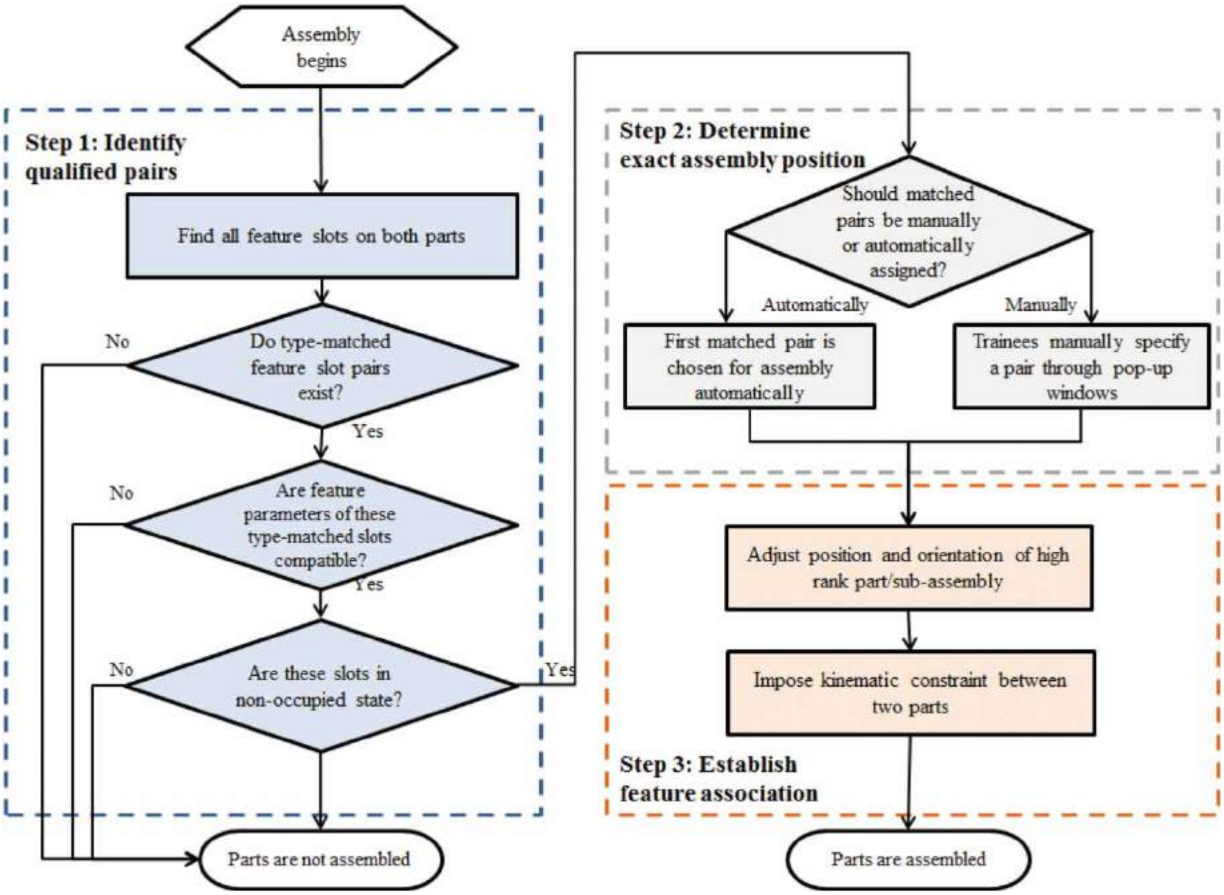

Fig. 10: Flowchart of establishing a feature association after collision of two mechanical parts.

7. EVALUATION OF VIRTUAL ASSEMBLY LABORATORY

The framework for assembly has been applied in developing a planet gear train laboratory that is a part of a junior-level mechanical engineering course on "Mechanisms and Machine Dynamics". In the development of the laboratory, the assembly rules described in section 4 and 5 were applied and all mechanical parts, such as planet gears, sun gears, ring gears, etc. described in that section were modeled and coded. Students can use the operations described in section 6 to assemble planet gear trains. By performing the laboratory exercises, students can enhance their understanding of planet gear trains.

In the VE laboratory, students are tasked to collaboratively build a planetary gear train. The laboratory provides three sets of sun gears (each with different dimensions), ring gears, planet gears and planet carriers for the students to choose from. Fig. 11 shows the steps of the assembly procedure and Fig. 12 depicts the selection panel in the VE. Among all 27 potential configurations, only three combinations of dimensions are valid for assembling a functioning planet gear train. These combinations are not given to the students. Instead, the students are required to find them on their own. This laboratory exercise was administered to 35 mechanical engineering students, who were divided into groups of 2 or 3 students each. The detailed procedure of this laboratory exercise and the learning effectiveness of the laboratory exercise were discussed in detail previously [9].

In this article, the focus is on the process of the students performing the assembly in the VE. From the students' activities in the laboratory, it was observed that all groups were able to complete the assembly (with the help of a teaching assistant). Also, it was noticed that most groups chose a suitable combination of dimensions by performing an analysis first and then fixed any possible analysis problems by experimenting in the VE, which is the procedure preferred by the instructor. If the students' analysis was correct, they were able to complete the assembly with one single trial. However, those groups who did not analyze the dimensions at all before attempting the assembly had to try several combinations in the VE. In the worst case, one student group could theoretically have to try as many as 27 different combinations in order to assemble a viable planet gear train. Tab. 1 illustrates the number of actual trials made by the students.

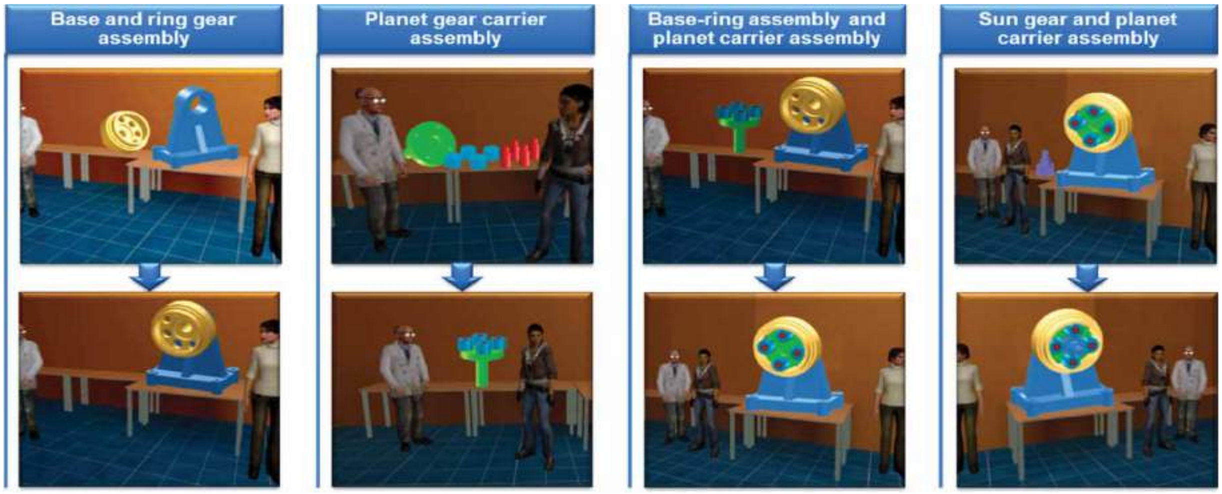

Fig. 11: Snapshots of assembly procedure for planetary gear system in VE.

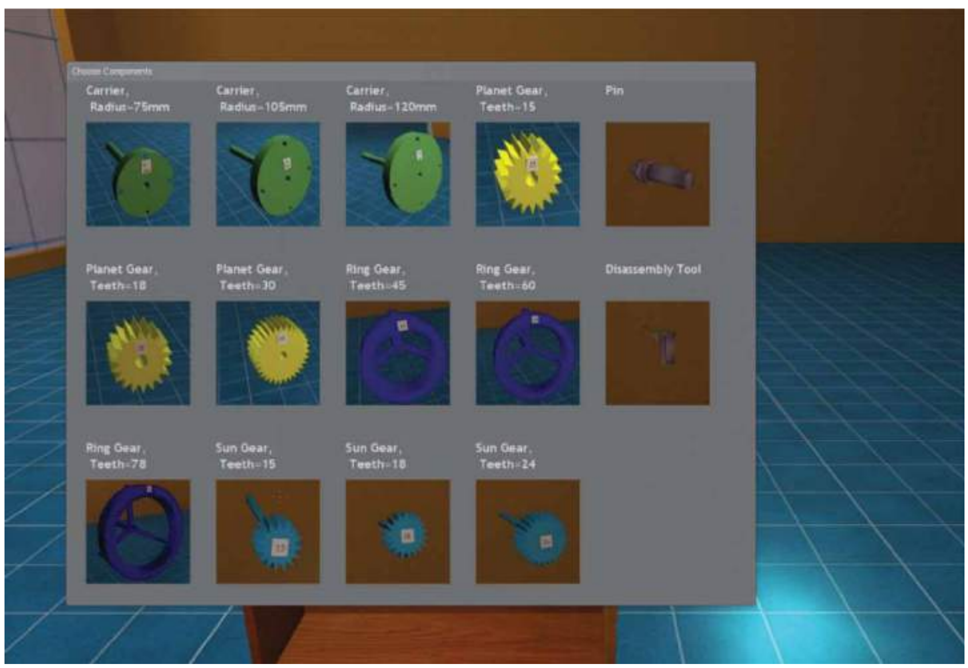

Fig. 12: Gear selection panel in VE laboratory.

Number of trials	*1*	*2*	*3*	*4*	*5*	*6*	*7*
Number of groups	6	1	2	2	2	0	2

Tab. 1: Number of groups versus number of trials.

From Tab. 1, it can be seen that among 15 groups, 6 groups (40%) were able to complete the assembly with one single trial. This fact shows that these students fully understood the gear train theory and performed a correct gear train analysis prior to assembling their design in the VE. Two groups tried 7 times, which is still much less than trying to assemble gear trains randomly. This indicates that they had done analysis before or during the assembly, and it also shows that through trials, they were able to gradually test combinations of dimensions in the VE, which eventually led them to a viable configuration.

8. CONCLUSIONS AND FUTURE RESEARCH

This paper introduces a framework for authoring VEs for mechanical assembly training. This framework is based on adapting a computer game engine originally designed for implementing entertainment environments for the development of educational laboratory environments. In order to make the game engine compatible with state-of-the-art engineering capabilities, a feature-based assembly representation was created. In accordance with the feature-based design concept, this representation includes assembly features, feature associations, kinematic constraints and subassembly hierarchies. The operations involved in an assembly process are discussed and an algorithm for judging the validity of such operations is described.

The main contribution of this work is that the framework presented here reduces the complexity and costs associated with developing virtual assembly training environments by using 3D game engine technology. This approach enables less experienced developers to design and implement similar applications. Even though game engines in general are not designed to support high levels of detail or dimensional accuracy, this framework expands the game engines' capabilities, thus allowing the implementation of mechanical assembly functionality. Furthermore, the approach described here enables developers of virtual environments to create game modifications (also known as 'Mods') that involve kinematic constraints between virtual objects without the need to access the game engine's source code. Therefore, this framework has the potential for being applied to industrial-level training and education.

As an example, a virtual laboratory for the assembly of gear trains was developed based on this framework. In this virtual laboratory, student groups are tasked with assembling planetary gear trains. They have to choose correct combinations of gear types and dimensions. A pilot study for this laboratory showed promising results. Although this was the first exposure of the students to such a virtual laboratory environment, they were able to complete the laboratory assignment.

Future work will investigate how to insert in real time more realistic mechanical models of real objects into VEs without requiring the creation and importing of CAD models. Furthermore, it would be desirable to devise methods for facilitating more efficient interactions between humans and virtual environments on the basis of technologies such as 3D sound synthesis, stereoscopic 3D displays and haptic feedback [42,43]. Voice recognition could also be used to facilitate the human-computer interaction.

ACKNOWLEDGMENT

This multi-disciplinary research project was carried out at Stevens Institute of Technology with funding from NSF CCLI Grant No. 0817463. This support is gratefully acknowledged.

REFERENCES

[1] Abouel Nasr, E. S.; El-Tamimi, A. M.; Abidi, M. E.; Al-Ahmari, M.: Virtual assembly in a semi-immersive environment, World Academy of Science, Engineering and Technology, 74, 2013, 2–26.

[2] Amory, A.: Game object model version II: a theoretical framework for educational game development, Educational Technology Research and Development, 55(1), 2007, 51–77. http://dx.doi.org/10.1007/s11423-006-9001-x

[3] Andreoli, R.; De Chiara, R.; Erra, U.; Scarano, V.: Interactive 3D environments by using videogame engines, Proceedings of Ninth International Conference on Information Visualization IV'05, London, UK, July 6-8, 2005.

[4] Annie Jin, S. A.; Bolebruch, J.: Avatar-based advertising in Second Life: the role of presence and attractiveness of virtual spokespersons, Journal of Interactive Advertising, 10(1), 2009, 51–60. http://dx.doi.org/10.1080/15252019.2009.10722162

[5] Ausburn, L. J.; Ausburn, F. B.: Desktop virtual reality: a powerful new technology for teaching and research in industrial teacher education, Journal of Industrial Teacher Education, 41(4), 2004, 1–16.

[6] Aziz, E.-S.; Chang, C.; Arango, F.; Esche, S. K.; Chassapis, C.: Linking computer game engines with remote experiments. Proceedings of the ASME International Mechanical Engineering Conference & Exposition IMECE'07, Seattle, Washington, USA, November 11–15, 2007.

[7] Aziz, E.-S.; Esche, S. K.; Chassapis, C.: An architecture for virtual laboratory experimentation, Proceedings of the 2006 ASEE Annual Conference and Exposition, Chicago, Illinois, USA, June 18–21, 2006.

[8] Aziz, E.-S.; Chang, Y.; Esche, S. K.; Chassapis, C.: Capturing assembly constraints of experimental setups in a virtual laboratory environment. Proceedings of the ASME International Mechanical Engineering Conference & Exposition IMECE'12, Houston, Texas, USA, November 9–15, 2012.

[9] Aziz, E.-S.; Corter, J.; Chang, Y.; Esche, S. K.; Chassapis, C.: Evaluation of the learning effectiveness of game-based and hands-on gear train laboratories, Proceedings of the Frontiers in Education Conference, Seattle, Washington, USA, October 3–6, 2012.

[10] Aziz, E.-S.; Esche, S. K.; Chassapis, C.: Design and implementation of a virtual laboratory for machine dynamics, International Journal of Online Engineering, 6(2), 2010, 15–24.

[11] Aziz, E.-S.; Chang, Y.; Tumkor, S.; Esche, S. K.; Chassapis, C.: Adopting computer game technology to support engineering laboratories, Proceedings of the ASME International Mechanical Engineering Conference & Exposition, IMECE'10, Vancouver, British Columbia, Canada, November 12-18, 2010.

[12] Bayarri, S.; Fernandez, M.; Perez, M.: Virtual reality for driving simulation, Communications of the ACM, 39(5), 1996, 72–76. http://dx.doi.org/10.1145/229459.229468

[13] Gupta, R.; Whitney, D.; Zeltzer, D.: Prototyping and design for assembly analysis using multimodal virtual environments, Computer Aided Design, 29(8), 1997, 585–597. http://dx.doi.org/10.1016/S0010-4485(96)00093-0
[14] Howard, B. M.; Vance, J. M.: Desktop haptic virtual assembly using physically based modeling, Virtual Reality, 11(4), 2007, 207–215. http://dx.doi.org/10.1007/s10055-007-0069-3
[15] Hu, W.; Zhang, X.: A rapid development method of virtual assembly experiments, International Conference on Electronic & Mechanical Engineering and Information Technology, 16, 2011, 39–68.
[16] Jayaram, S.; Connacher, H. I.; Lyons, K. W.: Virtual assembly using virtual reality techniques, Computer-Aided Design, 29(8), 1997, 575–584. http://dx.doi.org/10.1016/S0010-4485(96)00094-2
[17] Jia, R.; Xu, S.; Gao, S.; Aziz, E.-S.; Esche, S. K.; Chassapis, C.: A virtual laboratory on fluid mechanics, Proceedings of the 2006 ASEE Annual Conference and Exposition, Chicago, Illinois, USA, June 18–21, 2006.
[18] Jayaram, S.; Jayaram, U.; Kim, Y. J.; DeChenne, C.; Lyons, K. W.; Palmer, C.: Industry case studies in the use of immersive virtual assembly, Virtual Reality Journal, 11, 2007, 217–228. http://dx.doi.org/10.1007/s10055-007-0070-x
[19] Lee, C. H.; Liu, A.; Del Castillo, S.; Bowyer, M.; Alverson, D.; Muniz, G.; Caudell, T. P.: Towards an immersive virtual environment for medical team training, Studies in Health Technology and Informatics, 125, 2006, 274–279.
[20] Li, J. R.; Khoo, L. P.; Tor, S. B.: Desktop virtual reality for maintenance training: an object oriented prototype system (VREALISM), Computer in Industry, 52(2), 2003, 109–125. http://dx.doi.org/10.1016/S0166-3615(03)00103-9
[21] Marks, S.; Wünsche, B.; Windsor, J.: Enhancing virtual environment-based surgical teamwork training with non-verbal communication. Proceedings of the Fourth International Conference on Computer Graphics Theory and Applications, Lisboa, Portugal, February 5-8, 2009.
[22] Memi, E.: Boeing's working on augmented reality, which could change space training, ops. http://www.boeing.com/news/frontiers/archive/2006/october/i_ids03.pdf.
[23] Nickerson, J. V.; Corter, J. E.; Esche, S. K.; Chassapis, C.: A model for evaluating the effectiveness of remote engineering laboratories and simulations in education, Computers & Education - An International Journal, 49(3), 2007, 708–725.
[24] O'Neil, H. F.; Mayer, R. E.; Herl, H. E.; Niemi, C.; Olin, K.; Thurman, R. A.: Instructional strategies for virtual aviation training environments, In: Aircrew Training and Assessment, 2000, 105–130.
[25] Pausch, R.; Crea, T.; Conway, M.: A literature survey for virtual environments – military flight simulator visual systems and simulator sickness, Presence: Teleoperators and Virtual Environments, 1(3), 1992, 344–363.
[26] Rachuri, S.; Han, Y. H.; Feng, S. C.; Wang, F.; Sriram, R. D.; Lyons, K. W.; Roy, U.: Object-oriented representation of electro-mechanical assemblies using UML, Proceedings of the IEEE International Symposium on Assembly and Task Planning, 2003, 228–234. http://dx.doi.org/10.1109/ISATP.2003.1217216
[27] Rachuri, S.; Han, Y. H.; Foufou, S.; Feng, S. C.; Roy, U.; Wang, F.; Sriram, R. D.; Lyons, K. W.: A model for capturing product assembly information, Journal of Computing and Information Science in Engineering, 6(3), 2006, 11–21. http://dx.doi.org/10.1115/1.2164451
[28] Saaski, J.; Salonen, T.; Hakkarainen, M.; Siltanen, S.; Woodward, C.; Lempiäinen, J.: Integration of design and assembly using augmented reality, In Micro-Assembly Technologies and Applications, Springer US, 2008, 395–404.
[29] Shiratuddin, M. F.; Thabet, W.: Utilizing a 3D game engine to develop a virtual design review system, Journal of Information Technology in Construction, 2012, 267–271.
[30] Silva, R.; de Oliveira, J. C.; Giraldi, G. A.: Introduction to augmented reality, LNCC Research Report, No. 25/2003, National Laboratory for Scientific Computation, ISSN: 0101 6113.
[31] Smith, S. P.; Trenholme, D.: Rapid prototyping a virtual fire drill environment using computer game technology, Fire Safety Journal, 44(4), 2009, 559–569. http://dx.doi.org/10.1016/j.firesaf.2008.11.004
[32] Source SDK Programming: https://developer.valvesoftware.com/wiki/Category:Programming
[33] The top 10 elements of good game design: http://www.gamefaqs.com/features/top10/2449.html
[34] Thomsen, C. W.: Visionary Architecture: from Babylon to Virtual Reality, 1994, Prestel Verlag, Munich.
[35] Tumkor, S.; Aziz, E.-S.; Esche, S. K.; Chassapis, C.: Integration of augmented reality into the CAD process, Proceedings of the 2013 ASEE Annual Conference and Exposition, Atlanta, USA, June 23–26, 2013.
[36] VEGAS: A Virtual Environment for General Assembly Simulation. http://public.vrac.iastate.edu/jmvance/ASSEMBLY/Assembly.html.
[37] Villalta, M.; Gajardo, I.; Nussbaum, M.; Andreu, J. J.; Echeverría, A.; Plass, J. L.: Design guidelines for classroom multiplayer presential games (CMPG), Computers & Education, 57(3), 2011, 2039–2053. http://dx.doi.org/10.1016/j.compedu.2011.05.003

[38] Voxmap PointShell software: http://www.boeing.com/boeing/phantom/vps/.
[39] VR Juggler software: https://code.google.com/p/vrjuggler/.
[40] Walczak, K.; Cellary, W.; White, M.: Virtual museum exhibitions, Computer, 39(3), 2006, 93–95. http://dx.doi.org/10.1109/MC.2006.108
[41] White, M.; Mourkoussis, N.; Darcy, J.; Petridis, P.; Liarokapis, F.; Lister, P.; Walczak, K.; Wojciechowski, K.; Cellary, W.; Chmielewski, J.; Stawniak, M.; Wiza, W.; Patel, M.; Stevenson, J.; Manley, J.; Giorgini, F.; Sayd, P.; Gaspard, F.: ARCO – an architecture for digitization, management and presentation of virtual exhibitions. Proceedings of Computer Graphics International, 2004, 622–625.
[42] Xia, P. J.; Lopes, A. M.; Restivo, M. T.; Yao, Y.: A new type haptics-based virtual environment system for assembly training of complex products, International Journal of Advanced Manufacturing Technology, 58, 2012, 379–396. http://dx.doi.org/10.1007/s00170-011-3381-8
[43] Yoon, J.: Assembly simulations in virtual environments with optimized haptic path and sequence, Robotics and Computer-Integrated Manufacturing, 27(2), 2011, 306–317. http://dx.doi.org/10.1016/j.rcim.2010.07.015
[44] Zachmann, G.: Virtual Reality in Assembly Simulation: Collision Detection, Simulation Algorithms, and Interaction Techniques. Dissertation, Fraunhofer-IRB-Verlag, 2000.
[45] Zha, X. F.; Du, H.: A PDES/STEP-based model and system for concurrent integrated design and assembly planning, Computer-Aided Design Journal, 34(14), 2002, 1087–1110. http://dx.doi.org/10.1016/S0010-4485(01)00186-5
[46] 5DT: http://www.5dt.com/products/pdataglove5u.html.