Sacramento 1997
ASME Design for Manufacturing Symposium

Virtual Assembly with Construction Kits

Bernhard Jung, Martin Hoffhenke, Ipke Wachsmuth
Faculty of Technology, Knowledge-Based Systems Group
University of Bielefeld
{jung,martinh,ipke}@techfak.uni-bielefeld.de

15th January 1997

Abstract

The *CODY*[1] *Virtual Constructor* is a knowledge based system that enables an interactive assembly of 3D-computergraphically visualized parts to complex aggregates. The user interacts with the system either by direct manipulation or by issuing commands in simple natural language. This functionality is achieved by enriching the graphical data structures of the VR system with a layer of high-level, step-keepingly updated knowledge representations. For example, through the knowledge based description of the objects' connection ports the multi-functional parts from construction kits can be fitted together in a large variety of novel combinations. Thus, new virtual prototypes can be designed in the virtual environment using intuitive VR interaction means. Furthermore it is possible to define a structured model of a target aggregate against which the current state of the assembly is permanently matched. Therefore, progress of the assembly process is monitored and verbal instructions can always refer to the current state of the assembly. We have developed a knowledge representation formalism, COAR, as well as inference mechanisms over COAR representations that are especially tailored towards such assembly tasks in virtual environments.

---

[1] CODY is a project in the collaborative research center "Situated Artificial Communicators" (SFB 360) at the University of Bielefeld and is in part supported by the German National Science Foundation (DFG).

1 Introduction

Recent years have seen a number of advances in manufacturing technology promising cost-effective product design, manufacturing, and maintenance. In manufacturing, the use of standardized construction kits has become widespread. Construction kits provide a limited number of multi-functional parts which are reused in a variety of products. At the intersection of CAD modeling, Virtual Reality (VR) as well as other disciplines, Virtual Prototyping has emerged as a technology that aims at the construction of "digital mock-ups" instead of physical ones. Interactive examination of virtual prototypes by way of so-called "walk-throughs" were shown useful for easy visual verification of product design.

However, the design process of virtual prototypes typically still relies on conventional CAD modelling methods. I.e. the designer completely "assembles" the prototype within the CAD system by entering alpha-numerical values into appropriate fields of the system editor. Upon completion, the geometry data is exported to a separate VR system for inspection of the digital mock-up. If a construction fault is discovered the design-and-inspection cycle is started over again. A deficit of this approach is that no means are provided that allow the designer to change the assembly structure in the virtual environment directly. Thus, this methodology does not take advantage of new VR interaction means such as direct manipulation or natural language instructions to assemble the prototypes in the virtual environment.

In the following, we present an operational system, the *CODY Virtual Constructor*, that enables an interactive simulation of assembly processes with construction kits in a virtual environment. The user interacts with the system either by direct manipulation or by issuing commands in simple natural language. This functionality is achieved by enriching the graphical data structures of the VR system with a layer of high-level, step-keepingly updated knowledge representations. For example, through the knowledge based description of the objects' connection ports the visualized objects can be fitted together in a large variety of novel combinations. Thus, new virtual prototypes can be designed in the virtual environment using intuitive VR interaction means. Furthermore it is possible to define a structured model of a target aggregate against which the current state of the assembly is permanently matched. This allows the simulation of real assembly processes where the system is able to evaluate the progress and correctness of the assembly.

The paper is organized as follows: In Section 2, the system's interaction modalities and main action types are described. Section 3 describes the system components focussing on the central role of knowledge processing for virtual assembly. In Section 4, we conclude and discuss our results.

2 Virtual Assembly

With the CODY Virtual Constructor, novel mechanical assemblies can be created from the 3D-computergraphically visualized parts of construction kits. In our testbed scenario, the user can assemble a toy airplane and similar constructs from parts of the Baufix construction kit, such as bolts, blocks, and bars (see Figure 1). As in typical VR systems, the user can inspect the graphics scene from different perspectives, i.e. "walk through" the virtual scene. But, enhancing the functionality of standard VR systems, the user may also assemble complex aggregates from the blocks either by way of direct manipulation or by instructing the system in simple natural language. Main action types supported by the Virtual Constructor are assembly, rotation of sub-aggregates, and disassembly operations.

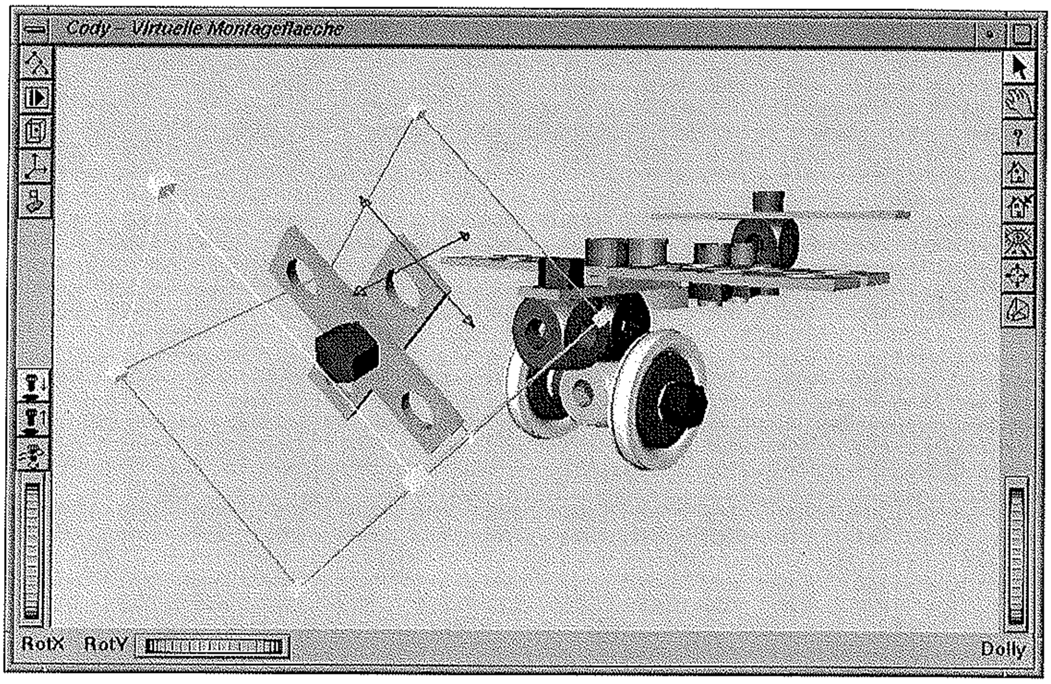

Figure 1: Sample assembly via direct manipulation: The propeller is dragged, e.g. using the mouse, towards the Baufix airplane's fuselage. When moved close enough, the objects snap together and the fitting process is completed.

One interaction modality provided the Virtual Constructor is the direct manipulation of parts using the mouse, space-mouse, data-glove or similar input devices. Single parts or whole aggregates can be picked and dragged across the virtual scene. In assembly mode, if the moved objects are brought in a position where one of their connection ports is close enough to the connection port of another object, a snapping mechanism will complete the fitting process. Figure 1 shows a snapshot of the Virtual Constructor's user interface. In the interaction shown, assembly mode is selected (indicated by the highlighted button on the left window frame) and the airplane's propeller is dragged by way of direct manipulation. The propeller can be attached to the aggregate in the back of the scene by fitting its bolt in any of the free holes of the aggregate's building blocks. Collision detection avoids the assembly of physically impossible constructs. Similarly, rotation and disassembly steps can be performed using direct manipulation only.

Sample assembly: *"Put the bolt from below into the middle hole of the bar"*.

Sample rotation: *"Rotate the lower bar crosswise to the tail unit"*.

Sample disassembly: *"Disconnect the right axle from the undercarriage"*.

Figure 2: Main action types supported by the Virtual Constructor and sample natural language instructions.

The other interaction modality provided by the CODY Virtual Constructor is the use of instructions in simple natural language. Figure 2 shows several examples, each depicting one of the main action types (assembly, rotation, disassembly) supported by the CODY Virtual Constructor.[2]

While both direct manipulation and natural language instructions are intuitive means for user interaction in virtual environments, we also found it useful to provide the user with audio feedback about the actions taken. Each action type is associated with a characteristic sound generated upon the successful completion of an action. For example, each assembly step is accompanied by a "clicking" sound. In case a user instruction cannot be completed, e.g. due to the user not anticipating an object collision resulting from an assembly instruction, a special sound is generated indicating failure. In this way, the user is always informed about the success or failure of intended assembly steps.

3 Knowledge Processing

In the CODY Virtual Constructor, the possibility of virtual assembly is achieved by a knowledge based, step-keepingly updated description of the virtual scene. In this section, we first give a raw overview of the Virtual Constructor's system architecture, emphasizing the role of the knowledge processing component in the overall system. We then describe the knowledge processing component in more detail, focussing on the knowledge representation language COAR, the inference mechanisms used for the step-keeping updating of COAR representations, as well as the interfacing of COAR representations with the graphical objects of the virtual scene.

3.1 System Overview

A (very) simplified overview of the Virtual Constructor's system architecture is presented in Figure 3. At the heart of this architecture is the knowledge processing component that maintains both geometrical and conceptual knowledge about the current state of the virtual scene. The following list describes how other system components draw on information provided by the integrated geometric and conceptual knowledge processing component:

- Natural language processing component: Dynamic knowledge about currently assembled aggregates is required, for example, to evaluate user instructions referring to composites (*"undercarriage"*, see Figure 2 bottom) or functional names of single parts (*"axle"*, see Figure 2 bottom).

---

[2] All natural language instructions in this paper are translations of original German input. An English natural language processing component is currently under development.

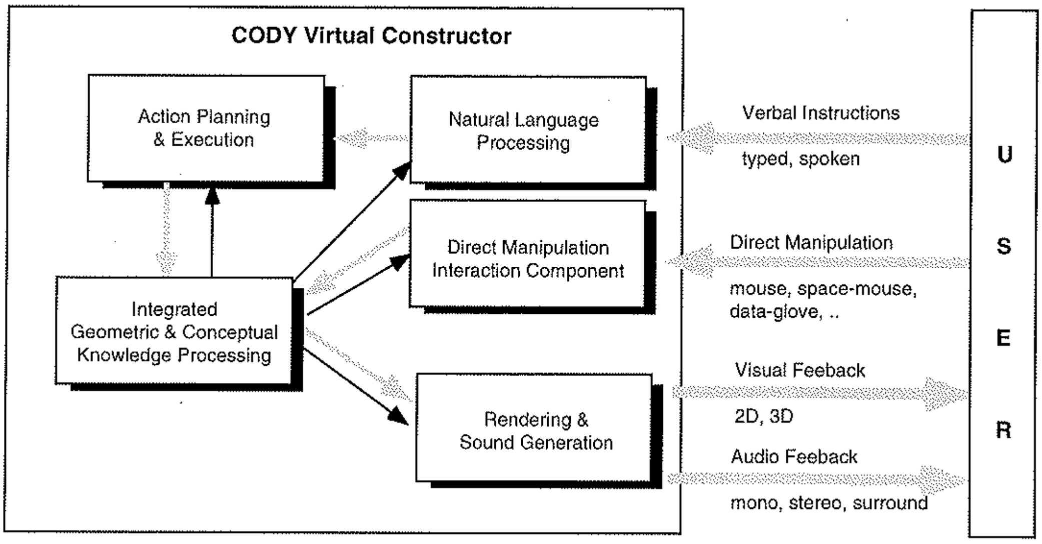

Figure 3: Simplified system architecture of CODY Virtual Constructor. Grey arrows indicate control flow, dark arrows indicate data flow from knowledge processing component to other system components.

- Action planning and execution component: Knowledge about the target position of moved objects is necessary, e.g., when executing assembly instructions (see Figure 2 top). Knowledge about current connection structure is required to achieve a realistic object behaviour in all kinds of actions (e.g. see Figure 2 middle: front bolt must be moved with the lower bar referenced in the instruction).

- Direct manipulation processing component: Analogously to action planning and execution component, knowledge about the current connection and aggregate structure is used to ensure realistic object behaviour. E.g. when dragging an aggregate across the virtual scene, all parts of the aggregate must be moved synchronously (see Figure 1).

- Rendering and sound generation component: Audio feedback is provided upon completion of all operations, e.g. clicking after assembly steps. Furthermore, highlighting of certain of certain objects and aggregates can be used to give the user visual feedback about the types and locations of aggregates assembled in the virtual scene; e.g. the instruction *"show all wheels"* highlights all wheels present in the virtual scene.

3.2 Integrated Geometric and Conceptual Knowledge Processing

The knowledge processing component of the CODY Virtual Constructor comprises and integrates several static and dynamic knowledge bases as well as inference methods. The static knowledge bases define generic knowledge about the objects and aggregates of the assembly task, while the dynamic knowledge bases contain current representations of objects and assemblies in the virtual scene. The inference methods achieve the updating of the dynamic knowledge bases. Furthermore, the knowledge processing component is divided into a geometric and a conceptual level. The geometric level, roughly, provides information corresponding to the graphical data bases in typical VR systems, such as position and shape information, whereas the conceptual level, roughly, contains additional information about the assembly structure of the visualized objects. An overview of the integrated geometric and conceptual knowledge processing component is given in Figure 4.

3.2.1 Conceptual Level

On the conceptual level of the knowledge processing component, two static knowledge bases define generic high-level knowledge about the objects and aggregates of the assembly task. One static knowledge base defines conceptual knowledge about the the mechanical objects of the construction kit, such as their type and connection ports. A second static knowledge base defines the target aggregate's structured assembly groups as well as the specific roles the multi-functional building blocks can assume in assembly groups. For example, a BLOCK assumes the role of an UNDERCARRIAGEBLOCK when it is used as component of a UNDERCARRIAGE.

Furthermore, two dynamic knowledge bases are maintained on the conceptual level containing what we call *dynamic conceptualizations* of the scene, i.e. dynamically updated conceptual representations describing the current situation in the (virtual) environment [WJ96]. In the first of the dynamic knowledge bases, the virtual scene is conceptualized w.r.t. the generic knowledge about the construction kit. Here, connected objects are straightforwardly grouped into "flat" aggregates, but not interpreted as subassemblies of the target assembly. The second of the dynamic knowledge bases contains the dynamic scene conceptualization w.r.t. the target assembly. Here, the aggregates of the virtual scene are interpreted as assembly groups of the target assembly.

The representations of both the static and dynamic knowledge bases are modeled in the representation formalism COAR (*"Concepts for Objects, As-*

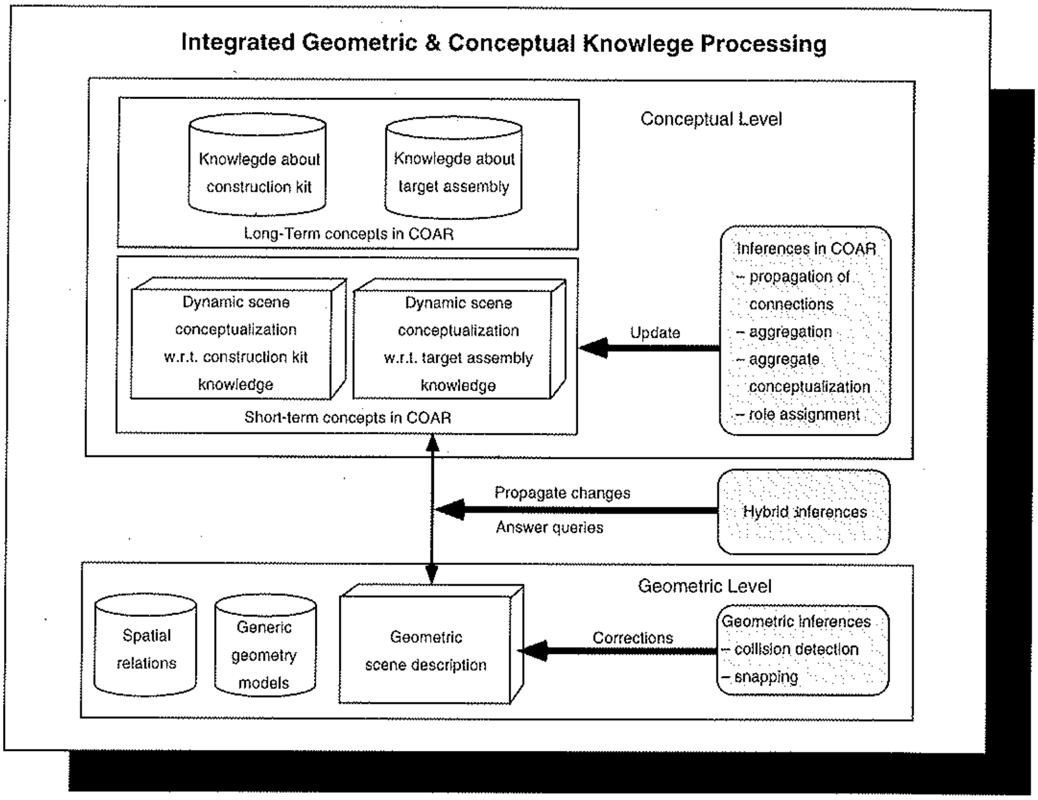

Figure 4: Integrated geometric and conceptual knowledge processing component: Static knowledge bases (cylinders) define generic knowledge, dynamic knowledge bases (cubes) contain current knowledge about the assembly task. Inference mechanism (grey) update the dynamic knowledge bases.

semblies, and Roles”) [Jun96, WJ96]. COAR is a frame-based language that supports the modeling of multi-functional mechanical objects, the assemblies that can be constructed from these objects, as well as the specific functional roles that objects can assume when used as part of larger assemblies. COAR builds on the semantic network language ERNEST [KNPS93], and the terminological language for part-of hierarchies introduced in [PL94], but is especially tailored towards the representation needs of on-going assembly tasks in virtual and real environments. In COAR, generic representations are called *long-term concepts*, current representations of individual objects (primitive and composite) are called *short-term concepts*. Long-term concepts for assembly groups are described by their supertypes, parts, part-part-(pp-)constraints describing necessary relations between their parts, as well as other attributes. COAR further divides long-term concepts into *object*

Long-Term Concept: UNDERCARRIAGE
is-a: ASSEMBLYGROUP
part has-left-halfaxlesystem: HALFAXLESYSTEM
part has-right-halfaxlesystem: HALFAXLESYSTEM
part has-block: UNDERCARRIAGEBLOCK
pp-constraint connection ⟨has-block⟩ ⟨has-left-halfaxlesystem⟩
pp-constraint connection ⟨has-block⟩ ⟨has-right-halfaxlesystem⟩
pp-constraint parallel$_x$ ⟨has-left-halfaxlesystem⟩ ⟨has-right-halfaxlesystem⟩

Figure 5: COAR definition of a long-term concept describing a generic Baufix undercarriage (explanation see text).

types describing multi-functional objects and *role types* describing specific functional roles of assembly components.

Figure 5 shows the COAR definition of a long-term concept describing a generic Baufix undercarriage. An undercarriage is a complex assembly group consisting of three parts: two half-axle-systems (which are complex assembly groups themselves) and one undercarriage-block (representing the role assumed by a Baufix block when used as component of a undercarriage). Furthermore, each of the half-axle-systems must be connected to the undercarriage-block and the two half-axle-system must be parallel$_x$ to each other. An example of an undercarriage is shown at the bottom left of Fig. 2, an example of a half-axle-system is shown at the bottom right of Fig. 2.

The following inference mechanisms have been defined over COAR representations that achieve the updating of short-term concepts according to changes in the virtual scene [Jun96]:

- Propagation of connections: Connection relations are established between composite objects provided that a connection relation has already been established between some of their parts.
- Aggregation: Connected objects are grouped together by the creation of new "flat" aggregate representations.
- Aggregate Conceptualization: "Flat" aggregates are matched against the structured model of the target assembly. New representations of assembly groups are created when the match is established.
- Role Assignment: Assembly components are reclassified w.r.t. their role type according to their functional role in the larger assembly. Also, new context-dependent features are attributed to components by this inference.

3.2.2 Geometric Level

On the geometric level, two static knowledge bases define generic knowledge about the mechanical objects' geometric properties. The first one contains the building blocks' generic geometry models. They define the building blocks' wire frame models, bounding boxes, center of gravity, and prototypical orientation. Furthermore, they define the relative positions and orientations of the objects' connection ports. A second static knowledge base defines several qualitative spatial relations over the geometry models, such as $\mathsf{parallel}_x$, $\mathsf{orthogonal}_z$, near and touches. Also, a connection relation is defined between the objects' connection ports. This connection relation is most crucial for virtual assembly, because it enables the system to automatically recognize the connection structure of the objects in the virtual scene.

A geometric scene description contains data about current object positions and orientations. Furthermore, geometric aggregate representations are dynamically created or deleted in the geometric scene description whenever aggregate representations are created or deleted on the conceptual level. The geometric aggregate representations are used to provide the natural language processing component of the CODY Virtual Constructor with spatial information, such as position and size, about assembled aggregates.

On the geometrical level, certain inferences are defined that can correct the objects' positions and orientations. These inferences include collision detection and a snapping mechanism used to finish "almost complete" fitting operations in a physically realistic manner when interaction mode is direct manipulation (see Fig. 1).

Finally, inferences must be provided that keep the different scene representations on the geometric and conceptual levels of the knowledge processing component synchronized. Every time, a new connection can be established on the geometric level, a connection relation is established between corresponding object representations on the conceptual level. In the other direction, when a new aggregate representation is created on the conceptual level, a corresponding geometric aggregate representation is created on the conceptual level. Finally, whenever the aggregate conceptualization inference requires the verification of a spatial relation, such as $\mathsf{parallel}_x$, between objects, this information will be provided from the geometric level.

4 Conclusion

We have described the CODY Virtual Constructor, a knowledge based system that enables the interactive construction of mechanical assemblies in a virtual environment. The Virtual Constructor especially enables the design of novel assemblies from multi-functional parts of construction kits. Direct manipulation and natural language instructions can be used to change the objects' configuration in the virtual scene.

The key for the enablement of virtual assembly is the knowledge-based description of the virtual scene. Dynamic knowledge representations must be step-keepingly created, modified, and deleted to reflect the assembly structure of the evolving aggregate. We have developed the knowledge representation language COAR and the necessary inference methods over COAR representations to meet these requirements.

Besides the wooden construction kit of our testbed scenario, We have also imported and successfully assembled industrial CAD models with the CODY Virtual Constructor. In comparison to the conventional design process of virtual prototypes using CAD-based methods, the CODY Virtual Constructor makes fuller use of the intuitive interaction means provided by current VR technology. Especially when virtual assemblies with standardized, reusable construction kits are considered, the extra effort for the required knowledge-based modeling is readily payed off.

References

[Jun96] B. Jung. *Wissensverarbeitung für Montageaufgaben in virtuellen und realen Umgebungen.* PhD thesis, University of Bielefeld, 1996.

[KNPS93] F. Kummert, H. Niemann, R. Prechtel, and G. Sagerer. Control and explanation in a signal understanding environment. *Signal Processing*, 32:111–145, 1993.

[PL94] L. Padgham and P. Lambrix. A framework for part-of hierarchies in terminological logics. In *Principles of Knowledge Representation and Reasoning*, pages 485–496. Morgan Kaufmann, San Francisco, CA, 1994.

[WJ96] I. Wachsmuth and B. Jung. Dynamic conceptualization in a mechanical-object assembly environment. *Artificial Intelligence Review*, 10(3-4):345–368, 1996.