DIVIPRO: Distributed Interactive VIrtual PROtotoyping

Mashhuda Glencross*, James Marsh, Jon Cook, Sylvain Daubrenet, Steve Pettifer and Roger Hubbold
Advanced Interfaces Group, Department of Computer Science,
University of Manchester, UK

1 Introduction

DIVIPRO is a prototype tool for simulating the assembly and disassembly of mechanical engineering components. It is aimed particularly at situations where the responsibility for decision making is shared between geographically dispersed design teams, and provides a collaborative environment in which different team members concurrently visualize and manipulate models. It is currently being evaluated using models from the aerospace and medical industries.

DIVIPRO includes a number of components required to work together in real time: a geometric kernel for importing STEP models and performing geometric queries; a choice of two geometric constraint engines (basic and advanced depending on the level of functionality required); a flexible body simulator for modeling deformation of objects such as plastic tubes and cable harnesses; and a haptic module which allows forces from collisions, constraints and flexible components to be experienced by the user through a PHANToM force-feedback device.

The DIVIPRO distribution architecture aims to maximize user responsiveness while maintaining consistency, and it is this aspect of the project that forms the basis of this sketch.

2 The DIVIPRO System

Typically distribution in commercial CAD systems is performed using technologies such as Microsoft's NetMeeting, which aim to synchronize the displays of multiple users. This prevents users having different viewpoints, or simultaneously interacting with models. Each user in the DIVIPRO system has a different viewpoint and may simultaneously interact with models. This interaction is mediated by a constraint engine, the job of which is to check for collisions and apply constrained motion.

Distribution is managed by utilizing the University of Manchester's DEVA framework [Pettifer et al. 2000]. To a first approximation DEVA has a client-server architecture. This simplifies the issues of synchronization since the definitive state of an entity is only stored in one location. The disadvantage of this approach is that when a user interacts with a model; a round trip from the client to the server has to be made in order to validate the interaction, checking for collisions and constraints, and to update the state of the entity. If the server and clients are all located on a fast network (eg. a company's internal LAN) this round trip may be acceptable. However, with DIVIPRO we are targeting geographically dispersed design teams - a trial DIVIPRO session involved users located in Germany, Spain and the UK connected together via the Internet.

Due to the large distances involved (there is a 400ms ping time from the UK to Spain) the lag experienced by the users begins to make a client-server architecture unusable. In contrast a peer-to-peer architecture which performs the collisions and constraints calculations locally at each client maximizes responsiveness but at the cost of increased complexity in synchronizing the multiple clients.

Rather than being a single process, however, the DEVA server consists of a number of separate processes, which may be distributed across the network. Conceptually the server processes are presented as a single logical server, with their physical location hidden from clients. At any one time an entity is managed by a single server process, however it can migrate between server processes effectively "moving" across the network. This migration process is hidden from clients via an automatically updated naming layer.

DIVIPRO employs this novel migration technique to move the constraint engine to a location on the network closest (in terms of latency) to the user that initiates an interaction with a model. This minimizes the round trip time required to validate an interaction, resulting in a faster update rate both for this user and other observers.

In practice the overhead of migrating the constraint engine entity is too great since the complex data structures required to efficiently detect collisions and constraints are time consuming to rebuild. This problem was overcome by having multiple constraint engines simultaneously present, and migrating a simple token which permits only one of the constraint engines to be active at any one time. The inactive constraint engines listen to validated updates from the active one, thereby maintaining synchronization.

If a second user simultaneously begins to interact with the model, then that user is forced to use an already active constraint engine, which may not be optimal. The architecture minimizes the response time for a single user interacting with a model, eg. in a scenario that involves turn taking, but also maintains consistency for multiple interacting users, such as in a shared assembly task. Other migration strategies are currently being investigated: if multiple users are interacting then a fairer solution may be to migrate the constraint engine entity to a location equidistant from the interacting clients, or possibly to allow multiple constraint engines to be simultaneously active provided the objects the users are manipulating are sufficiently distant from each other that they do not interact.

The authors would like to thank the European Commission for providing funding and the following members of the DIVIPRO consortium: Labein, Sener, KL Tecknik, DLR, Eurocopter, and the University of Salford.

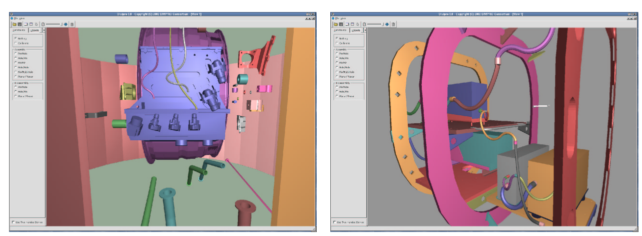

Figure 1: Screen shots showing the DIVIPRO system being used to assemble complex models.

Further information about DIVIPRO and other research projects at Manchester can be found at http://aig.cs.man.ac.uk/research/.

References

PETTIFER, S., COOK, J., MARSH, J., AND WEST, A. 2000. Deva3: Architecture for a large scale virtual reality system. In *Proceedings of ACM Symposium in Virtual Reality Software and Technology*, ACM Press, 33–39.

*e-mail: khotem@cs.man.ac.uk