Interactive Problem Solving in an Intelligent Virtual Environment

Carlos Calderon
University of Teesside
Middlesbrough, TS1 3BA, UK
c.p.calderon@tees.ac.uk

Marc Cavazza
University of Teesside
Middlesbrough, TS1 3BA, UK
m.o.cavazza@tees.ac.uk

Daniel Diaz
University of Paris 1
Paris, 75013, France
Daniel.Diaz@univ-paris1.fr

Categories and Subject Descriptors
D.3.2 [**Language Classification**]: Constraint and logic languages.
I. 2.8 [**Problem Solving**]: Graph and Tree search strategies

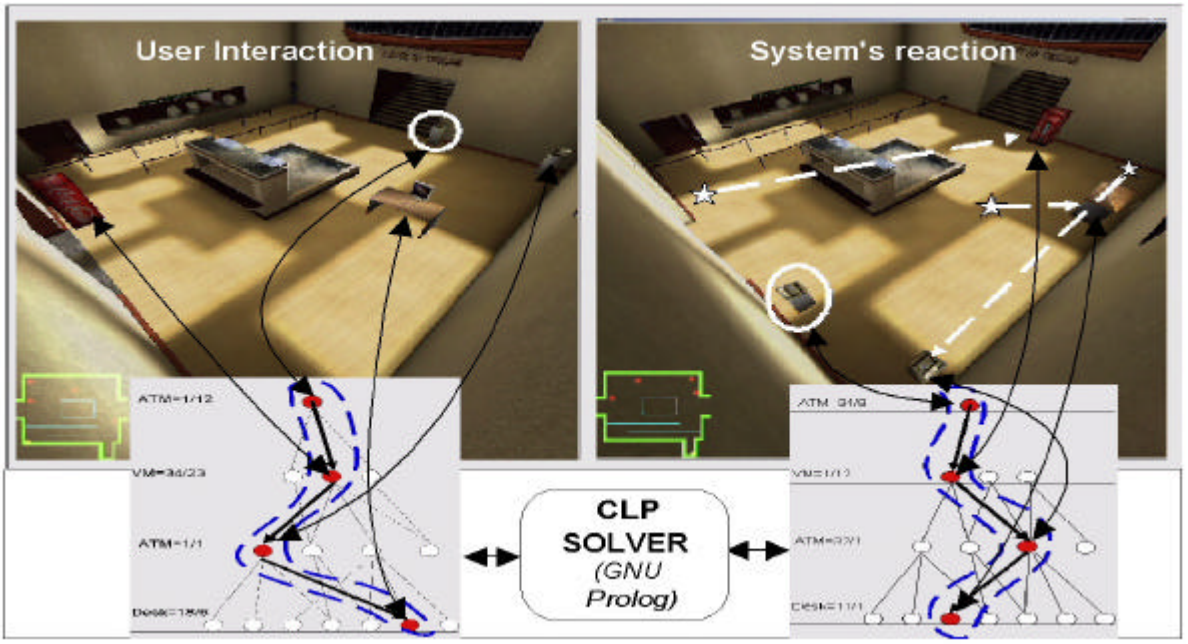
General Terms
Design, Languages, Verification

Keywords
Virtual Reality, intelligent interfaces, constraint programming

Virtual Environments (VE) can be used to represent complex situations. It is thus a logical evolution to extend them to carry out problem solving tasks as well. Intelligent Virtual Environments (IVE) integrate AI techniques with real-time 3D environments. As such, they can support interactive problem solving, provided the underlying AI techniques can produce solutions within a time frame matching that of the user's interaction. In these systems, the visual space is directly interfaced to a problem solver. Consequently, interacting with objects in the VE serves as input to the problem solver, which outputs new solutions directly as object configurations. This kind of system has many potential applications in design, configuration, situation assessment, etc.

We have implemented an intelligent virtual environment based on constraint logic programming (GNU Prolog), integrated in a real-time 3D graphic environment (Unreal Tournament™) using an event-based approach. More specifically, physical interaction with the virtual world objects can be translated into real-time input to the CLP solver by selecting the variables whose values have been altered by the interaction. For instance, when visualising a configuration, the user can alter the position of certain objects. This modifies the constraints involving these objects, triggering the solver on a new set of variables. Seen from the user's perspective, the 3D environment automatically reconfigures itself by reacting to his/her interactions. This form of interactive problem solving is illustrated in the figure below.

We demonstrate the system behaviour on a configuration example, which uses real-world design knowledge in terms of building interior design for offices (a bank agency in our case). In this system, the domain knowledge is expressed in the form of constraints in the object's attributes, which correspond not only to geometric constraints but also semantic constraints (such as compatibility between materials, lightning, etc.) .The system will propose a first result configuration, which can be interactively refined by the user, through direct physical modification in the virtual world, triggering the computation of a new solution compatible with the user's input. Our results show that the sampling rate of object manipulation in the virtual environment is compatible with the result production of the solver (on average, the total time is less than 0.15 seconds), hence supporting interactive exploration of the solution space.



IUI '03, January 12-15, 2003, Miami, Florida.