

## Computational Steering

R. van Liere, J.J. van Wijk

### 1. INTRODUCTION

The standard cycle in simulation is to prepare input, execute a simulation, and to visualize the results. Performing these activities simultaneously realizes greater insights and higher productivity. This is the underlying idea of *Computational Steering*: researchers change parameters of their simulation on the fly and immediately receive feedback on the effect. The development of a dedicated user-interface for a simulation is a time-consuming process, which requires the expertise of specialists on user-interfacing and computer graphics. Hence, CWI's Computational Steering project aims to develop an environment in which researchers themselves can develop and use interfaces to their simulations. The project generates considerable synergy between the system developers and researchers in a wide variety of disciplines. The project's output consists of research results in scientific visualization, tools such as CSE (Computational Steering Environment), and several applications in collaboration with researchers. The latter include a study of the atmosphere of the planet Venus (astrophysics, Vrije Universiteit Amsterdam, see below), energy research in connection with, e.g., ceramic foam burners, windmills, and high energy physics (Netherlands Energy Research Foundation ECN), and ball-bearings (the Swedish firm SKF).

## 2. BACKGROUND

Scientific Visualization has been a separate research area since the publication of an influential report by the US National Science Foundation in 1987 [1]. Many new methods, techniques, and packages have been developed over the past several years; but most of these are confined to post-processing of data-sets. The usual assumption is that all data is generated first, after which the researcher iterates through the remaining steps of the visualization pipeline (selection, filtering, mapping, and rendering) to achieve insight in the generated data. Hence, there is only limited interaction with the simulation.

*Tracking* is the first step to increasing interaction with the simulation. After each time-step of the simulation the resulting data for that time-step is sent into the visualization pipeline and can be inspected. If the researcher considers the results invalid, the simulation can be stopped at an early stage, and restarted with a different set of input parameters. The next step, *Computational Steering*, goes a lot further, and can be considered as the ultimate goal of interactive computing. Computational steering enables the researcher to change parameters of the simulation while the simulation is progressing.

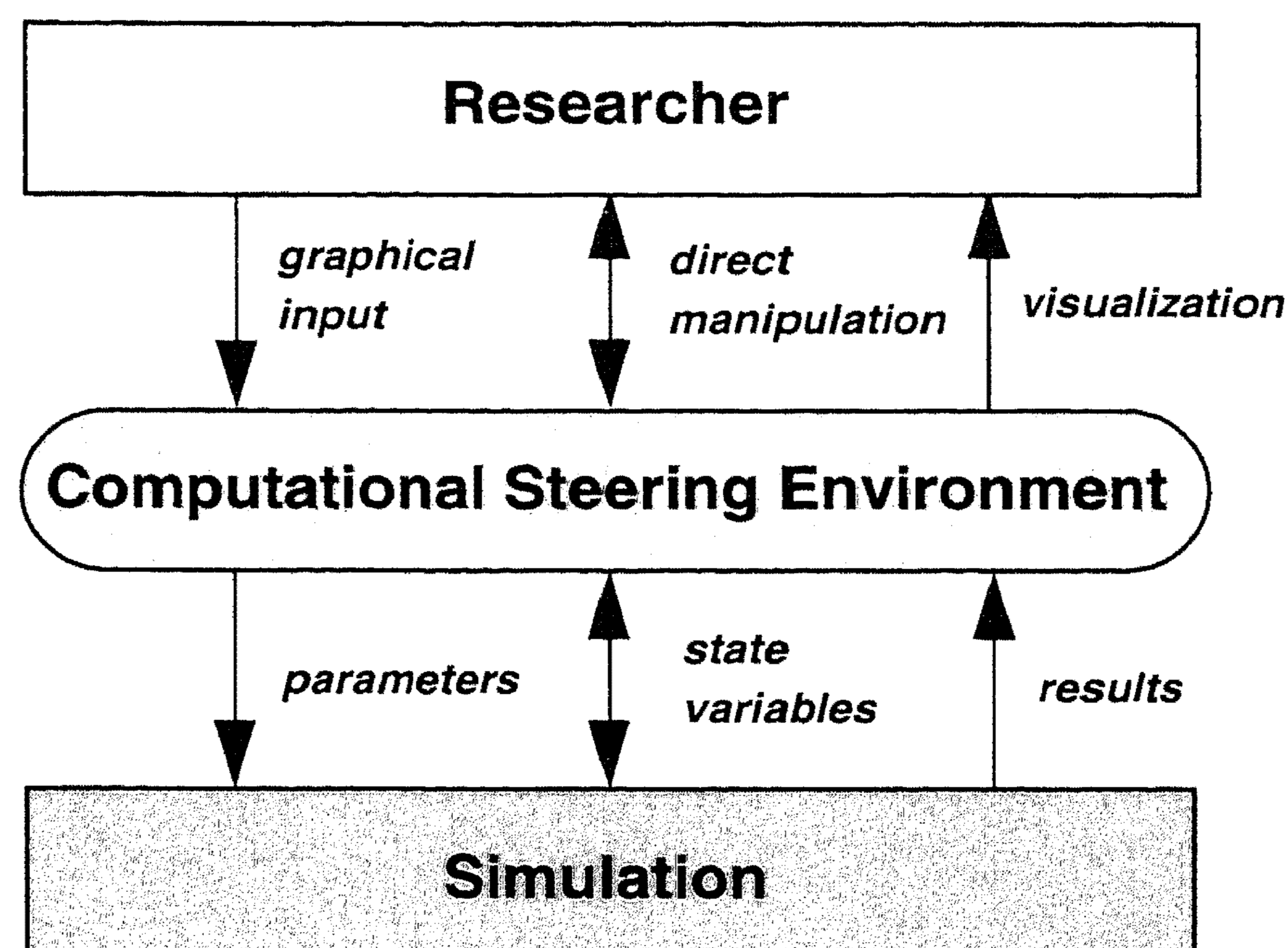
As an example, R.E. Marshall of the Ohio Supercomputer Center has applied computational steering to the study of a 3D turbulence model of Lake Erie. His conclusions were that: 'Interaction with the computational model and the resulting graphics display is fundamental in scientific visualization. Steering enhances productivity by greatly reducing the time between changes to model parameters and the viewing of the results.'

## 3. REQUIREMENTS

Although computational steering is an attractive concept, implementation is cumbersome. A researcher will need to ask a specialist in user-interfaces and visualization to develop a suitable tool for analyzing the simulations' output data. And in the weeks or months that takes, the chances are high that researcher's interests have shifted. Moreover, the use of computational steering will introduce new research questions, which prompt modifications of the tool. Hence, researcher and specialist will need to collaborate for an extended period.

The Computational Steering project started up in 1992 as a joint project between CWI and the Netherlands Energy Research Foundation ECN. The aim is to develop insights, methods, techniques, and tools that enable researchers to apply computational steering. The current focus is on development of a Computational Steering Environment (CSE) that encourages exploratory investigation by the researcher of a simulation.

What are the requirements for such an environment? The first set of requirements follows directly from the definition of computational steering.



**Figure 1.** Data flow between researcher, CSE, and simulation.

Researchers must be enabled to change parameters and to visualize results. The simulation must be enabled to read these parameters and to write the results. Certain variables fall both in the input and output categories. For instance, in a time-dependent simulation the state variables are calculated via integration, but occasionally the user might want to change the current value (position, velocity, etc.). This implies that the CSE must support direct manipulation: the user can interact with graphics objects that are also updated by the simulation. (See also figure 1.)

Second, the environment should enable researchers to use computational steering without help from visualization experts. In the process of gaining insight via computational steering, a researcher typically wants to look at and to control other, possibly new, variables, and to visualize them in various ways. This implies that it should be easy to define and refine an interface, as well as to connect a simulation with the environment, and to control variables through the interface.

#### 4. ARCHITECTURE

Given the set of requirements, how can a solution be realized? The architecture of the environment is centered around a data manager acting as a blackboard for communicating values, and satellites that produce and visualize data [3]. The purpose of the data manager is twofold. First, it manages a database of variables. Satellites can create, open, close, read, and write variables. For each variable the data manager stores a name, type, and value. Variables can be scalars or arrays, in which case the number and

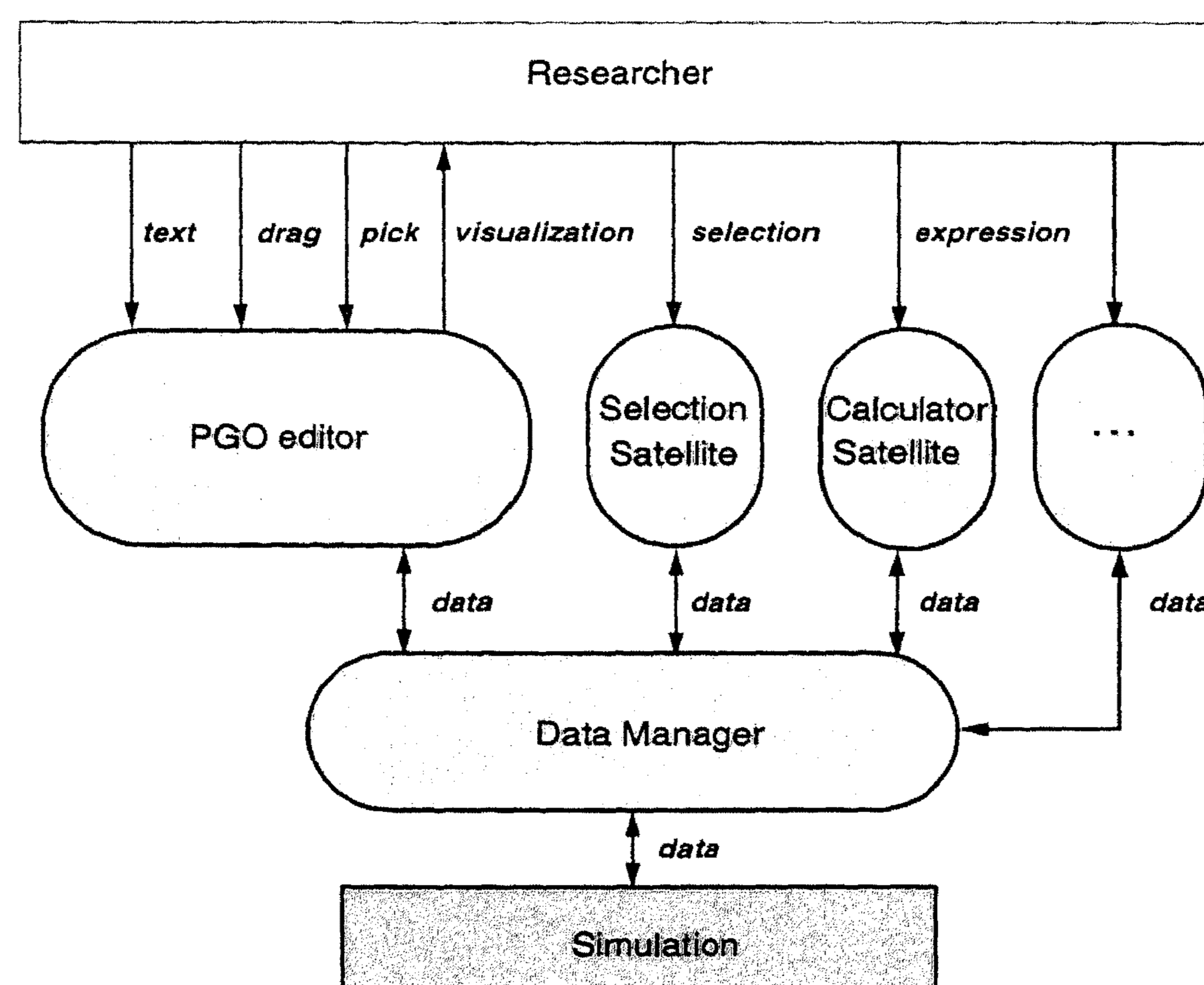


Figure 2. The CSE architecture.

size of the dimensions is also stored. Array sizes can change dynamically during the lifetime of the variable. Second, the data manager acts as an event notification manager. Satellites can subscribe to events that represent state changes in the data manager. (See also figure 2.) Whenever such an event occurs, the satellite will receive an event from the data manager. For example, if a satellite subscribes to mutation events on a particular variable, the data manager will send a notification to that satellite whenever the value of the variable is mutated.

High level libraries enable access to application interfaces to this environment. Simulations can easily connect with the data manager by simply declaring relevant variables.

390

A large suite of general purpose satellites is available for standard visualization tasks. For example, data can be logged, sliced, transformed, and calculated. However, the most important satellite is the PGO editor, a general purpose graphics editor for input and visualization of data.

##### 5. PARAMETERIZED GRAPHICS OBJECTS

The PGO editor is an interactive graphics, MacDraw-like, editing tool [2]. The central concept for the graphics editor is the Parameterized Graphics Object (PGO) : an interface is built up from graphics objects whose properties are functions of data in the data manager. Users sketch an interface and bind the graphics objects to variables by parameterizing geometry and attributes with data in the data manager. Simulations may drive the interface by mutating the data bound to the graphics objects. Similarly,

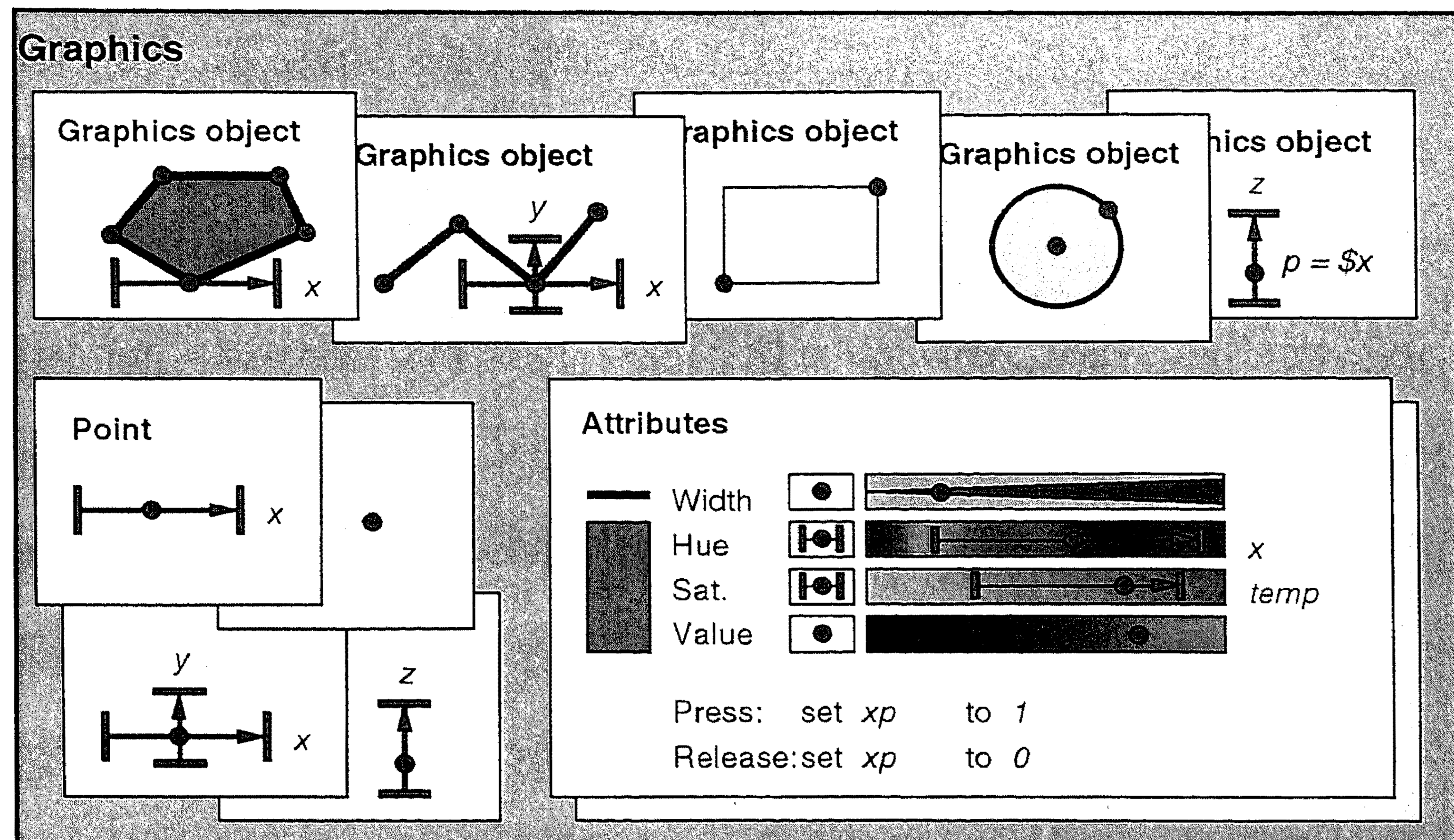


Figure 3. Parameterized Graphics Objects.

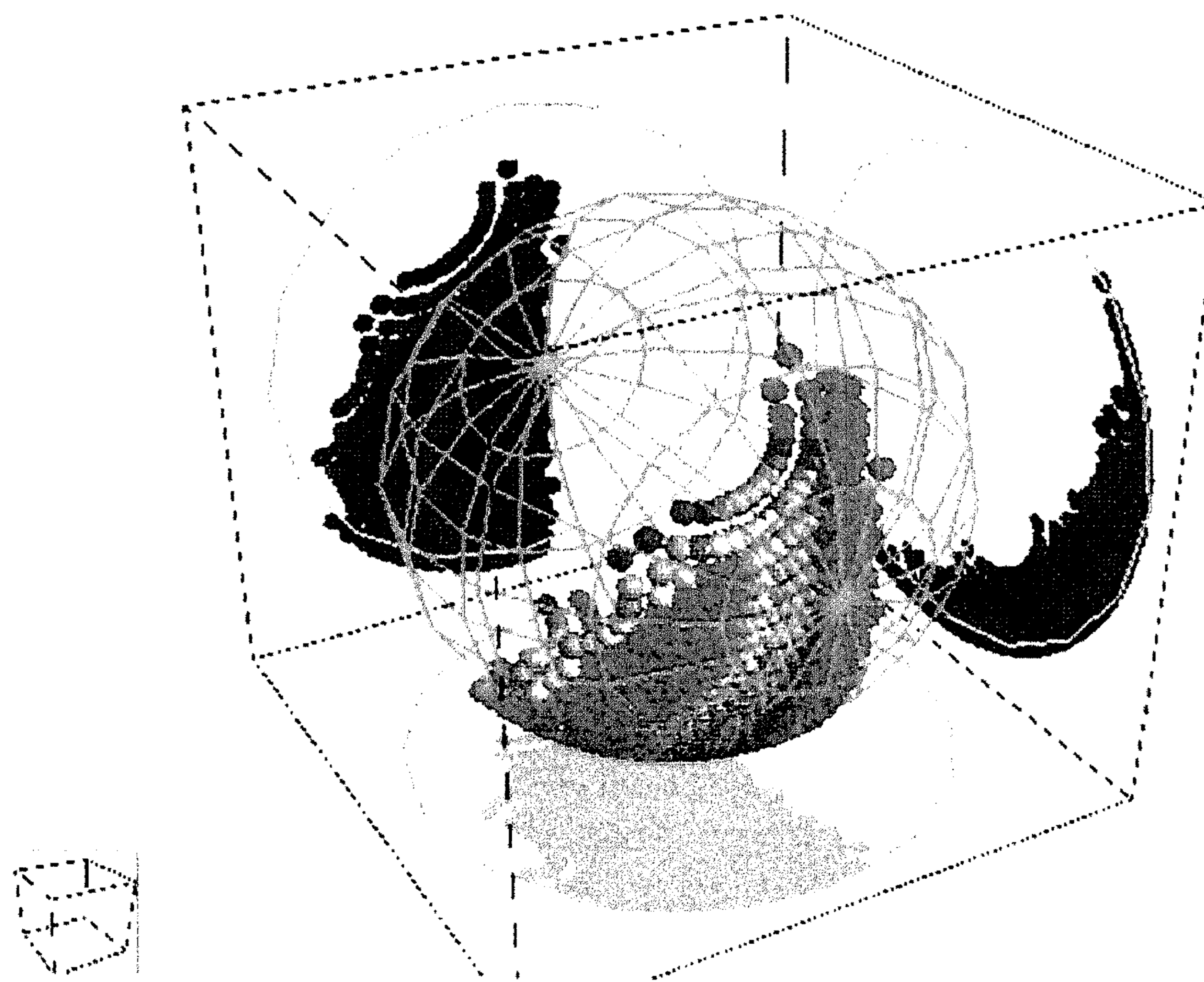
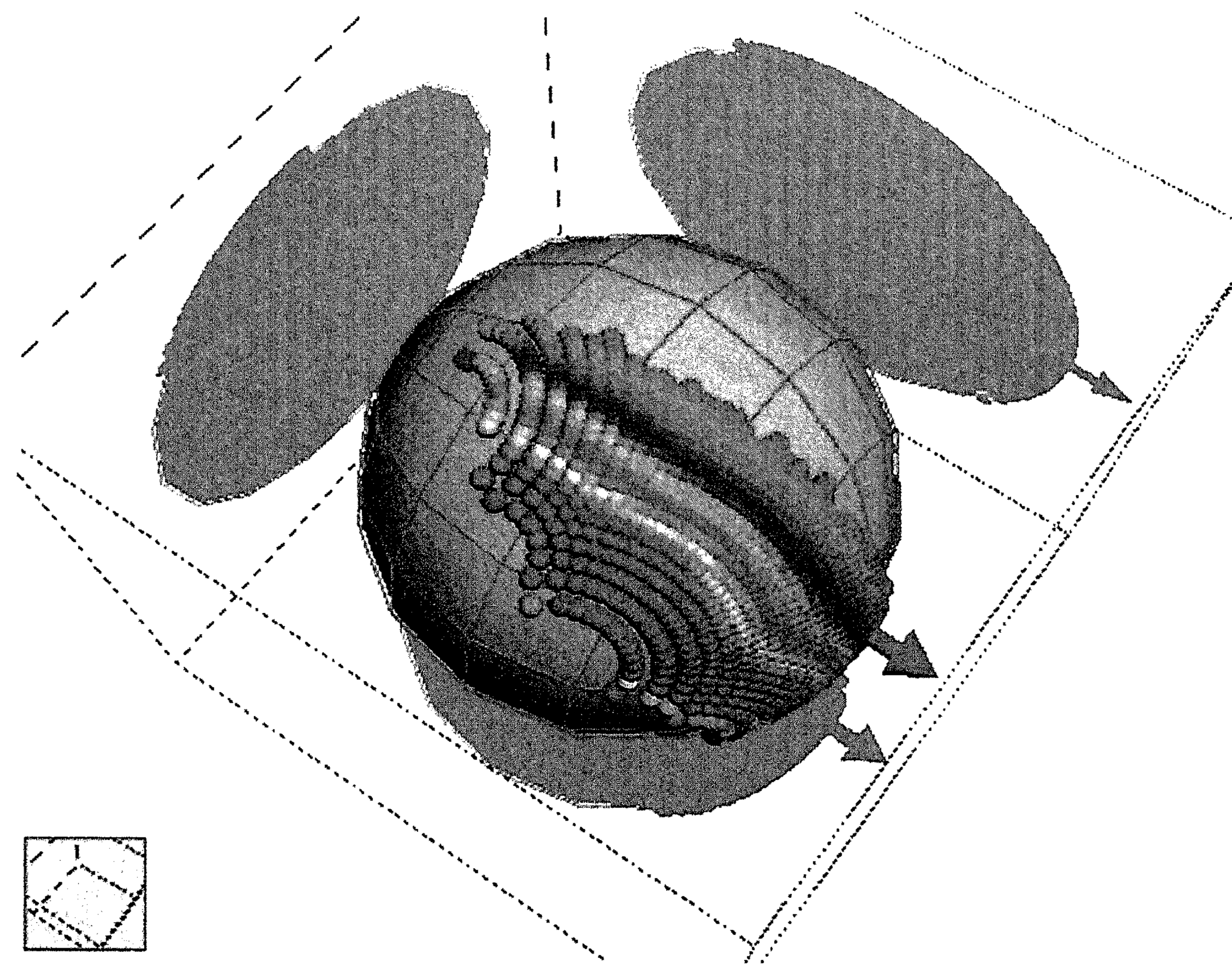
users may drive the simulation by interacting with graphics objects. Hence, a two-way communication between graphics and data in the simulation is supported. Since the PGO satellite is an interactive graphics editor, users may incrementally define the interface or change bindings. (See also figure 3.)

The graphics editor has two modes: specification and application, or shorter, *edit* and *run*. In edit-mode, the researcher can create and edit graphics objects. The geometry of the objects is defined by points and degrees of freedom. Degrees of freedom are parameterized to values of variables in the data manager. Furthermore, attributes of objects, such as color and linewidth, can also be parameterized.

In run-mode, a two-way communication is established between the researcher and the simulation. Data is retrieved from the data manager and mapped onto the properties of the graphics objects. Text can be entered, and objects can be dragged and picked, which is translated into changes of the values of variables in the data manager. Hence, there is automatic support for the usually time-consuming and error-prone direct manipulation of graphics objects.

## 6. APPLICATION

Several applications are underway in the Computational Steering project. Here we briefly describe one of them. Researchers at the department of



**Figure 4.** Two visualization snapshots, using 3D-PGO, of intensity measurements on the surface of the planet Venus, made during one day by the Pioneer satellite. The balls represent the degree of polarization.

astrophysics of the Vrije Universiteit study the structure of the atmosphere of Venus (e.g. to determine the vertical and horizontal size distribution of Sulphuric Acid droplets in the atmosphere) by comparing their computational models with measurements taken by the Pioneer satellite. Pioneer traces the atmosphere of Venus and takes measurements of polarization of the reflected sun light which is related to the structure of atmosphere. However, comparison between the measurements and the computational models is made very difficult because they are not independent: the measurements are used by the models to estimate model parameters in one dimension (a particular wavelength) prior to comparison of results in another dimension (another wavelength).

Lacking appropriate tools, the VU researchers were forced to reduce their original goal of modelling the complete Venus atmosphere to finding those points on the planet where the model parameter values are closest to the measurements and determine those values. Off-the-shelf software tools have enabled them to make many temporary and small conclusions based on calculations at relatively few points in the atmosphere.

Researchers at CWI are applying general computational steering concepts to provide more global and better insight to this application. Figure 4 shows CSE snapshots of the visualization of the intensity measurements from the Pioneer.

The CSE allows direct manipulation of modelling/simulation parameters at interactive speeds. While this type of interaction is a very important capability, current research is focussed on the development of higher level input and navigation techniques. These techniques will be developed and packaged as satellites.

#### REFERENCES

1. B. MCCORMIC, T. DEFANTI, M. BROWN (1987). Visualization in scientific computing. *Computer Graphics 22(6)*, (SIGGRAPH '88), 103-111.
2. J. MULDER, J.J. VAN WIJK (1995). 3D computational steering with parameterized graphics objects. *Proceedings Visualization '95*, IEEE Computer Society Press, Los Alamitos, CA.
3. J.J. VAN WIJK, R. VAN LIERE (1994). *An Environment for Computational Steering*, CWI Report CS-R9448, CWI, Amsterdam. *Proceedings of the Dagstuhl Seminar on Scientific Visualization*, 23-27 May 1994, Germany.