(iii) a data store, or access to a data store, for storing task definitions including time data indicating task execution times,

said module further comprising scheduling means for storing data selected from at least one of said task definitions, including said time data for the respective task definition or definitions.

This scheduling means can be used by the software system for allocating tasks amongst a plurality of software modules during control, monitoring and/or management of a process or apparatus.

The scheduling means for one software module may store data from more than one task definition, ordering the data so as to determine the order in which, in use of the system, the software module will control or carry out the relevant task.

Advantageously, the scheduling means may store the task data together with an indicator of status selected from at least two alternative statuses such as "tentative" and "firm". The indicator of status may be used by the scheduler to determine modes of managing such data. For instance, the scheduler may operate a time-out in relation to task data having the status "tentative", after which the data is deleted or can be overwritten by subsequent incoming data.

Particularly advantageously, the scheduling means may overbook resources by storing data from more than one task definition, said data storing overlapping time contraints.

According to a fourth aspect of the present invention, there is provided a visualisation arrangement for use in a software system for controlling, monitoring or managing a process or arrangement, said software system comprising a plurality of software modules provided with means to communicate with each other, wherein the visualisation arrangement comprises means to store and provide for display communication instances, or data relating thereto, occurring in relation to a single, selected software module, and means to store and provide for display communication instances, or data relating thereto, occurring between at least two of said software modules.

A debugging arrangement which allows the user to choose to review communications relevant either to a single software module, or to a community of communicating software modules, or to both, can offer a very effective debugging mechanism to the user.