Aurora is a sophisticated scheduling system that combines a variety of scheduling techniques, intelligent conflict resolution, and decision support to make scheduling faster and easier. The software's scheduling decisions take into account resource requirements, a variety of constraints, and any pertinent domain knowledge.

Once Aurora has created a schedule, it is displayed in a series of graphical displays that allow the user to see the resource allocations and the temporal relationships among the elements. These displays also allow the user to edit the schedule directly and easily.

As well as allowing rapid, easy schedule development, Aurora addresses issues overlooked by many scheduling systems.

Aurora focuses on resource requirements and temporal scheduling in combination. Considering these different scheduling aspects in combination is especially important in domains such as class and curriculum planning, where there are a range of resource requirements, the time frame may be rather flexible but there are a number of temporal constraints, and each improvement in the schedule translates to a large pecuniary advantage for the client.

Aurora was also designed to allow extensive customization. Many systems are incapable of taking the client’s special needs into account, and so many of the benefits of an automated system are lost. Needs and priorities vary widely from one company to another, even within a single domain, and the scheduler needs to be able to reflect that. This customizability also allows the program to take expert domain knowledge into account, because this knowledge can easily be encoded into the heuristics that the system relies upon to make its decisions.

In all cases the system assumes that the user knows best. It respects all changes he makes to the schedule, maintaining the resource allocations and temporal placement that he specified.
Aurora provides a number of unique features designed to ease the user’s work, increase the expressiveness possible in the schedule, and allow a greater degree of control over the results.

**Resource Sets** - Group related resources into different sets to take advantage of all useful attributes.

**Resource Requirements** - Associate resource requirements with both resources and activities to control their needs and preferences.

**Constraints** - Define temporal constraints, resource constraints, and spatial constraints to regulate the relationships among the scheduled elements.

**Freezing** - Specify elements that should not be moved, either individually or by specifying the frozen time window.

**Scheduling Trade-off Regulation** - Control the schedule quality / schedule time trade-off by dictating the amount of time the scheduler should put into finding the best possible schedule.

**Reports** - Create reports of resource usage that can then be loaded into a standard spreadsheet.

**User-Defined Slots** - Create additional element properties as needed, either for your own reference or in combination with software customization for easily extended functionality.

**Calendars** - Associate a calendar with an activity or resource to dictate its standard schedule, and any exceptions that schedule might have. These may include yearly holidays or one-time events.

**Hierarchical Relationships** - Edit element groups quickly and efficiently by taking advantage of the hierarchical element structure.

The resulting schedule may be viewed either by resource or temporal relationships. The results can be manipulated directly on the display.

- Update the schedule quickly and easily by graphically editing it from the display using standard drag and drop functionality, or click on elements to make broader editing changes.
- Expand the schedule easily by creating new activities and resources from the schedule display.
- Add comments and make graphical additions with the markup utilities.
- View all conflicts, and mark any “false” conflicts to be ignored.
- Print the schedule results, or export them to a GIF for easy distribution or inclusion in other documents.