Attribute- a specification over a service (time, price etc.)

First time when a user enters the site:

* The user providing a minimal amount of information (preferences or searching filters).
* The system converts these preferences into a user model
* The CCA adds default preferences to the user’s expressed preferences, generating a user model that is likely to reflect more accurately the user’s true preferences (*moderate price sensitivity, preference for fewer stops, and a weak preference for flying on fewer different airlines. These preferences are common to almost all users, though the system can revise them if the user’s preferences are atypical.*).
* **The system** finds flights that satisfy the given preferences, groups the flights into trips, and **ranks the trips** using the preferences in the user model. Of the top-ranked trips, three significantly different, undominated trips will be displayed along with two extrema, the cheapest trip and best non-stop trip.
* Extrema can elicit more information from the user about the relative weighting of their preferences and provide the user with critical information about how much a potential solution could be improved in terms of a specific attribute, given the available solutions and the current preferences

Building the user model

* The user model consists of a set of **constraints** and a **weighting** indicating the importance of each constraint.
  + We use the convention that when a constraint of a domain (of a set or tuple of attributes for each service e.g. flight, hotel booking etc.) is equal to 0 then the constraint (constraint it’s the user’s search preferences) is fully satisfied, and when is 1 fully unsatisfied.
  + The user itself can **add a new constraint, modify an existing constraint, or adjust the weighting of a constraint.**

**(***This can be accomplished, as in our implementation, though the use of a graphical user interface that allows the user to critique the solutions suggested by the CCA. After the user critiques the suggested candidates, CCA is called with the updated user model, which results in a new set of solutions being suggested***)**

After the system has built a user model:

Input/output for the CCA.

CCA takes as input all available solutions and a
partial user model and generates suggested solutions for the user to evaluate.


* The CCA is a function from a user model and a set of solutions to a small set (a constant number) of suggested solutions
* Apply improvements:
  + CCA will exclude dominated solutions (The CCA should never suggests a solution that is dominated by, or strictly inferior to, another suggested solution)
  + Prefer significantly different solutions (The CCA will not suggest solutions that are too similar to other suggested solutions)
* After a set of solutions has been suggested the user can either:
  + choose one and end the interaction
  + add a new constraint, modify an existing constraint, or adjust the weighting of a constraint.
    - How is adjusted the weighting of a constraint?

This can be accomplished using a graphical user interface that allows the user to critique the solutions suggested by the CCA. After the user critiques the suggested candidates, CCA is called with the updated user model, which results in a new set of solutions being suggested.