

# 1. Components of the DSS

## 1. List of components

Main approach separates algorithms from knowledge. Separate expert knowledge from fact knowledge.

- Knowledge Base: Is the storing of all kinds of knowledge that GGD have/is using. The following kind knowledge can be accessed by DSS internal services:
  - Case based facts: This is all kind of data that is available from the green house side. This data could be current or historical data. The DSS knowledge base is not managing, storing this data ! It is providing a connector to other services which are responsible for this data.
  - Expert Knowledge: Expert knowledge are the rules that are given by the expert or learned by the system. The GGD-DSS knowledge base is responsible for managing the rules. Additionally a list of all possible goals are also managed by the GGD-DSS
  - Inferred knowledge: Inferred knowledge is fact knowledge that is representing the result of applying the rules to the existing facts. Inferred knowledge must be managed, stored by other services outside of the GGD-DSS.
  - Inferred Goals: are the goals that are trained by the algorithm. Inferred goals must be stored and compared with the active goal.
- Rule engine: Is the logical component that applies the rules (expert knowledge) on the facts.
- A snapshot service : Is the component that is allowing the system to search back in time for the purpose of learning. This is achieved by storing the data periodically on the server with a timestamp.
- Rule editor: Is the component that allows to add, delete, update rules which are managed by the knowledge base. The rule editor is a program that runs on a client machine and can connect to a server to load from or update rules to.
- Learning engine: Is the component that is taking historical data and tries to identify rules autonomously out of this data. To make the DSS more efficient.
- A goal manager: The goal manager is the connection of the inferred goals of the knowledge base to other services.

## b. Interfaces offered to other services

- Goal management
  - Providing of all active goals with descriptions and parameters
  - Providing of all active goals with descriptions and parameters for a specific subsystem (bed)
  - Acceptance of user input: goal has been reached
  - Providing of explanation why goal is active
  - Providing of information since when goal is active
  - Provide list of all possible goals and possible values of parameters.
  - Change priority of a goal (just once or every time)
- Rule management
  - Providing all active rules (for a specific subsystem)
  - Activate / deactivate set of behaviours for a specific subsystem.
  - Give explanation to a specific rule. (first step this is the comment that we have stored for a rule and in a later step it is why the system has learned this rule)

o

### c. Version roadmap

- Integration point 1: First version: rule engine & knowledge base with goal manager
- Integration point 2: improved rule engine (able to handle historical data and why questions) & knowledge base with goal manager + rule editor
- Integration point 3: Learning engine - first Version and needed knowledge base support

## 2. Dependencies - Components needed by the DSS

Must:

- A service that is providing a storage for current facts - these facts could be data from sensors, data provided by the user (values - the user as a sensor),
  - o Interfaces:
    - Update a fact
    - Store new fact
    - Search for facts, objects or statements that are described by not bounded variables. (example give me all Sensor that are measuring the wetness in bed 2. provide an answer as a list of the query).
    - The service should be able to store who/which instants set a value or made a statement - then we can answer questions of why. Why do you think that the plant is dead? - Because rule r1 has said this. Why has rule r1 said that the plant is dead? Because the leaves are gray and the ground is dry? Why is the ground dry? \_Because sensor 123 has said this. (Quite powerful ;-)
    - Provide a transcript of expert actions with a list of keywords, since this can lead to setting a value or making a statement to the knowledge.
      - When did the expert perform an action?
      - Why did the expert perform an action?
      - How did he perform?
- A service that is providing historical data - values with time stamp, user input with time stamp, events with time stamp.....
  - o Store new entry: events, statements, change etc.
  - o Search for entries in a specific time frame etc.
  - o The service should be able to store who set a value or made a statement - then we can answer questions of why.

Optional:

- A service that connects to the DSS and taking the active goals to work with it. A service that is using the DSS like a user interface (manager).
- Design of a use case scenario by the expert