Modelling languages

What is a modelling language?

File formats

Standardized way to store and exchange data. Often not human readable. (rtf, jpeg, …)

Modelling/specification/description languages

Usually human writable/readable format for describing something with a complex structure. (query, structured data, graphic, circuit, OO program, optimization problem, rule, process,…)

Characteristics of modelling languages

Mix of data format, programming language constructs and built in procedures and algorithms to serve a particular type of application.

Iteration can be used to:

- control the sequential execution of built in procedures that are to be executed directly to produce some result. Eg. SQL database query language, Postscript page description language, ... (PROCEDURAL)

- make the description more concise. Processing the description will produce some intermediate model, description or code that can be used, analyzed or executed in various ways. E.g. AMPL math programming language, VHDL hardware design language. (DECLARATIVE)

Also *graphical* languages…

Graphical formalisms for e.g. object interaction, can also be viewed as modeling languages. Example UML, Petri nets, ...

Also belief networks for probabilistic expert systems, ...

(some sort of progression: graph, network, graphical language)

Advantages of modelling languages

Convenient special syntax and commands to define complex models.

Specialized functions can be performed, such as solving, verifying, draw image, ...

Models can be defined and exchanged in a standard format.

Note: the modelling language is usually not the model or model type itself, but rather a convenient way to describe models of a certain kind.

AMPL

Model: an LP or ILP problem

The language: quite straightforward constructions for constants, variables and constraints.

CLIPS rule based expert system shell

Double purpose with exercise:

- shows how knowledge is modelled in a rule based expert system
- a special CLIPS language

Mix of logical inference and action control (a so called production system). There are other kinds of systems working only with logical inference(=logical equation solving).

Based on production rules (in the code written with =>):

```
IF certain conditions are true
THEN execute the following actions
```

Possible actions include inserting and deleting facts from the knowledge base, input output etc.

CLIPS language details

Procedural semantics (forward chaining algorithm):
Check fact list with current facts.
Rules for which the conditions hold are put on the agenda.
Execute rule with highest salience(default 0), remove from agenda.
Repeat.
Stop if no rules on agenda.

LISP-like syntax.
assert: add fact
deffacts: add facts
retract: remove fact
?color: variable
?: wildcard in pattern matching
deffunction: do instructions in sequence