A goal/scenario-driven strategy to achieve fine granularity customization of software systems(**)

Julio Cesar Sampaio do Prado Leite(*)
Department of Computer Science, UofT
joint work with John Mylopoulos

talk at: University of Waterloo

*on leave from Departamento de Informática, PUC-Rio
** initial work

Summary

- What?
- Why?
- How?
- Challenges

What

- Variability has been an issue in Reuse, OO, and on Product Lines.
- Customization has been attracting RE attention recently (Fickas, Berry, Mylopoulos, van Lawsweerde).
- Personalization will require not only an effort in eliciting specific needs, but also will require ways of making possible the reuse of previous solutions, hence variability.
- Customization by goal modeling generates a huge space of choices (order 10 to the 10^{th} on the Oregon case study), thus a high variability.
- Research challenge: better methods, techniques and tools (from the point of view of SE) to deal with fine grain customization and high variability.

Why

- Future systems will network inexpensive cooperative processors to provide a distributed and powerful automation of living and working environments.
- One special type of application that will run on those systems are the ones designed to help impaired people.
- Being more specific: smart homes and offices.
- These systems will need application software with a high level of variability to handle personal customization.

How

- First of all, and once more: this is work in progress (initial stage).
- Many sources of ideas and results
  - Main:Hui, Liaskos and Mylopoulos
  - Others:
    - The TBI case study (e-mail) at Oregon
    - Lamsweerde: Goal and Obstacle analysis
    - Dagstuhl Product Family Workshop: Variability
    - L'Ecritoire (Rolland): goals and scenarios
    - Scenarios: PUC-Rio
- Departing from a published and well explored case study.

How

Hui, Liaskos and Mylopoulos

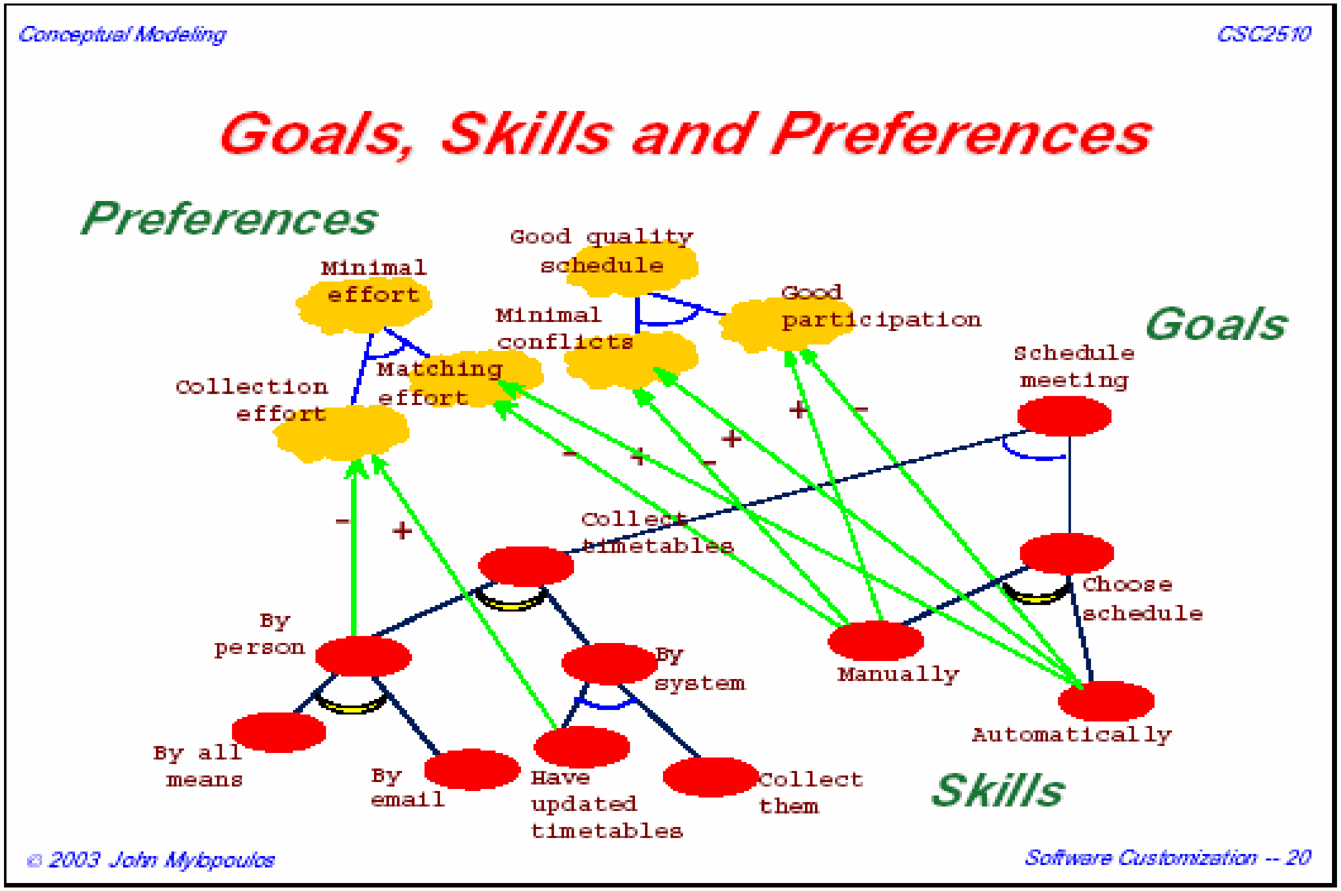
Our Proposal: Goals, Skills and Preferences

- *Gather **requirements** for the generic software system. Represent these as goals. The variability space is defined by the set of all possible ways one can satisfy these goals. Each alternative assigns **tasks** to users of the system.*
- *Identify **skills** required for a user to carry out a task that is needed for the fulfillment of a goal. Disallow alternatives that assign tasks to users who don't have the necessary skills.*
- *Represent user **preferences** as softgoals and use them to prioritize among alternatives.*
- *The **variability space** of the software is defined in terms of a goal model G. From the goal model we derive a feature model.*
- ***Customization** is defined as a mapping*

  Cust: G x S x P --> V

How

Hui, Liaskos and Mylopoulos



How

Hui, Liaskos and Mylopoulos

Goal Model Analysis

Using such goal models, we can answer questions such as:

- *What is the space of alternatives supported by the generic design?*
  …for our example, 6;
- *Rank alternatives with respect to a softgoal*
  …for our example,
  ***Alternative A:** system collects timetable constraints and schedules the meeting*
  ***Alternative B:** people do these tasks*
  A is better than B with respect to the "Minimal effort" softgoal;
- *Given a goal, find all alternatives that do/don't require certain skills.*

To support these types of analysis, we need formal models of goals, skills and preferences.

How

The TBI case study (e-mail) at Oregon

- Computer user profile
- E-mail task assessment
- Environmental and capabilities self-assessment
- User requirements (goals and expectations)
- Observation of natural communication and activity patterns and physical environment
- Monitoring needs
- Training plans

"Our initial approach was to attempt to use CORE to produce a fully-tailored system that was an exact match to the survivor's needs. In retrospect, this was the wrong path:…We now have what appears a more tenable approach. We use CORE to set up a training plan and to choose among a small set of small set of predefined, beginning systems."

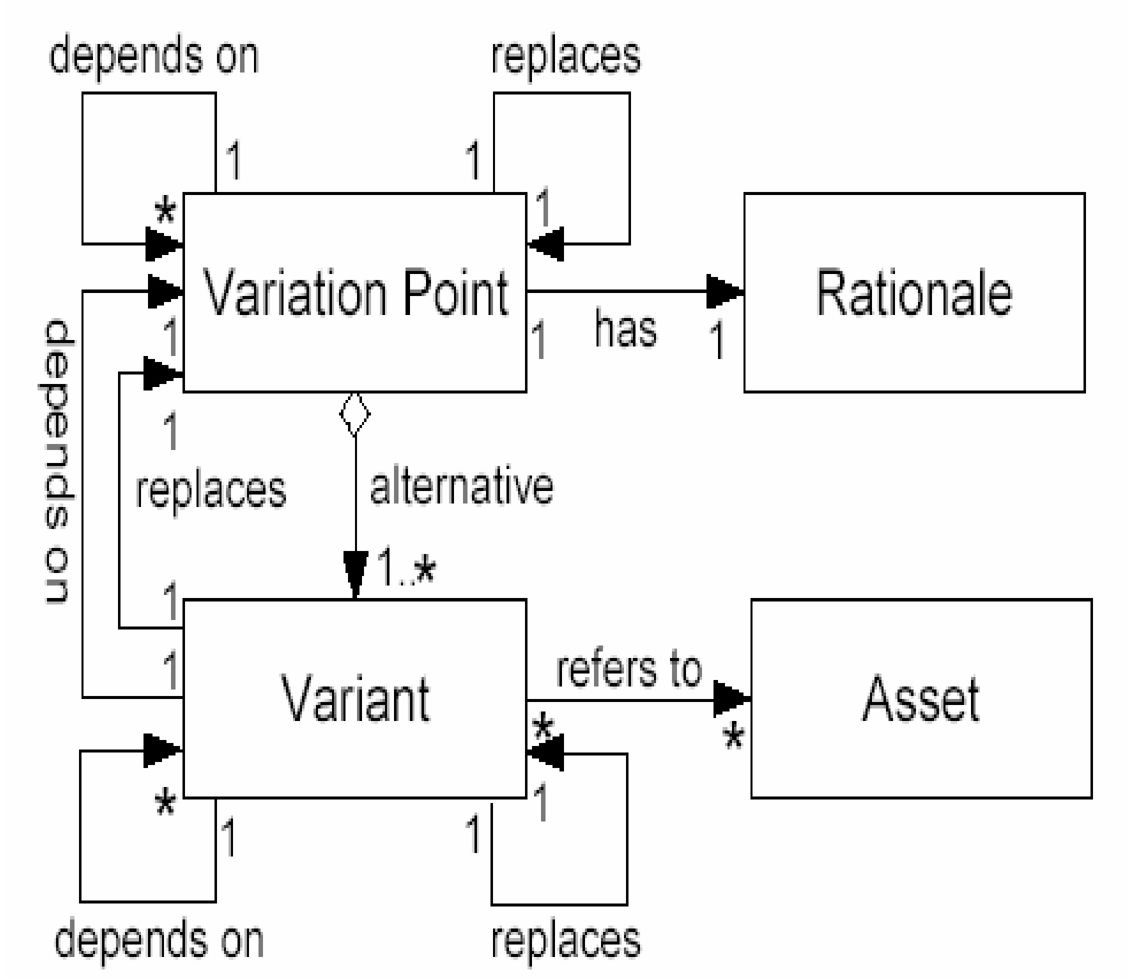
How

Lamsweerde et al. (TBI)

- Goal analysis
  - identify goals and alternatives
  - domain model
  - first version: including customizability NFG
- Obstacle analysis
  - generate obstacles (based on DM)
  - identify alternative of obstacles
- Customization analysis
  - develop-time
  - run-time

How

Dagstuhl Working Group



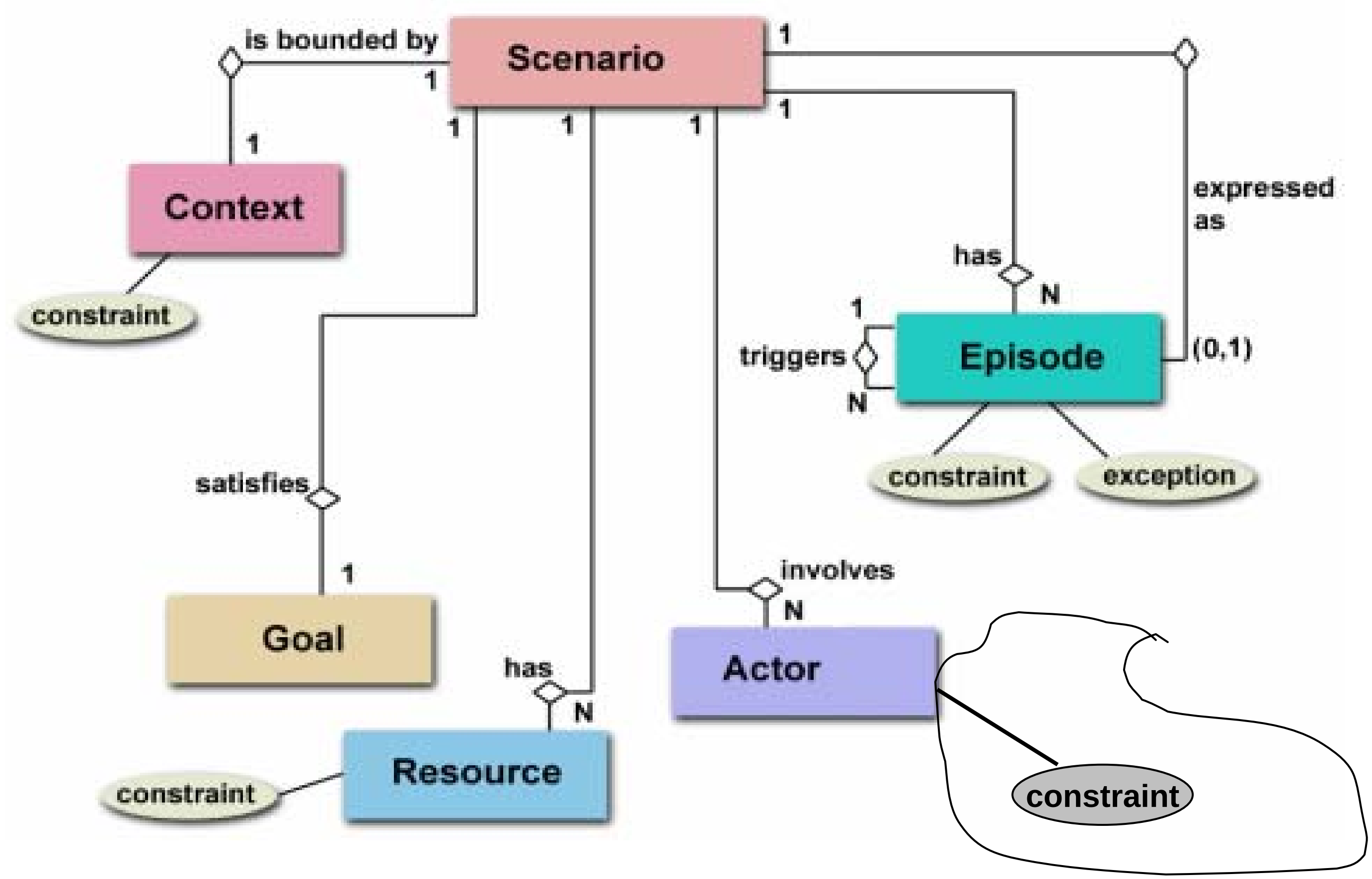
How

L'Ecritoire (Rolland): goals and scenarios

- Uses goal-scenario coupling (bi-directional)
- Analysis of textual scenarios
- Requirements chunk (<goal,scenario>)
- Goals are intentional
- Scenarios are operational
- From goals to scenarios: for each goal discovered a scenario is authored for it.
- From scenarios to goals: once a scenario is authored it is analyzed to yield goals.
- Naming conventions

How

Scenarios: PUC-Rio, UB, Tandil and LaPlata



How

Scenarios: PUC-Rio, UB, Tandil and LaPlata (needs a lexicon)

Scenario:Reoccupied *room*

Goal:Return to the previous *light scene*

Context: *Motion detector* is working, value T1 is known for this *room*. **Constraint:** T1 is less than 5 seconds. Any *room* in the *4th floor of building 32* .

Resources: *ceiling light groups* , *task lights* , *push-buttons* **Constraint:** soft buttons, *control panel*

Actors: *user*, *control system*, *motion detector* , *dimmer actuators* , *status lines*, *control system* ,

Episodes:

user enters the *room*.

motion detector signals to *control system* ***Constraint***:*: in less than one second.*

system verifies how long the *room* has been empty

If time has been shorter than T1 the *system* retrieves the last *chosen light scene* **Exception**: OCCUPIED *ROOM*

control system terminates the *standard light scene*

system implements last *chosen light scene* **Exception**: *Light* MALFUNCTION

How

Lexicon: PUC-Rio

Actuators / Actuator / Physical actuator

Notion:

It is a device that can be controlled by *control system* .

An *actuator* has name, abbreviation, *type*, *range*, control, *reaction time* and a description.

Behavioral responses:

It is controlled by the *control system* .

An *actuator* responds in linear time.
It controls *light*.

Sensors / Physical sensor / Sensor

Notion:

A device that can sense state of the building, users or environment.

A *sensor* has name, abbreviation, *type*, *resolution*, *range*, *reaction time* and *conversion time* .

Behavioral responses:

Analog sensors respond in exponential time.

A *sensor* is triggered by a physical occurrence under its *range*. (It is not defined
in the text, here is my definition)

Reaction time

Notion:

For a *sensor* , it is the time from a change of the sensed property to the time when the *sensor* has reached 90% of the change, excluding *conversion time*.

For an *actuator* , it is the time to change from 0 to 100% / 100 to 0%, if different.

Behavioral responses:

It is not defined in the text.

It is activated by a change in the environment (this need to be elaborated) in the case *o* f a *sensor*.

It is activated by the *control system* in the case of an *actuator*.

Dimmer actuators / lle / Dimmer actuator

Notion:

It is an *actuator*.

It controls the output of a *luminaire*.

Behavioral responses:

It is used to dim individually *ceiling light groups*

How

Our main idea (initial)

- Add variability to the Goal, Skill and Preference, but maintain the style of goal decomposition
- Initial solution: add variability at the scenario level and allow soft-goals to drive selection according to scenario constraints.
- Actual solution: add variability at three levels and allow soft-goals to drive selection.

How

Our main idea (initial)

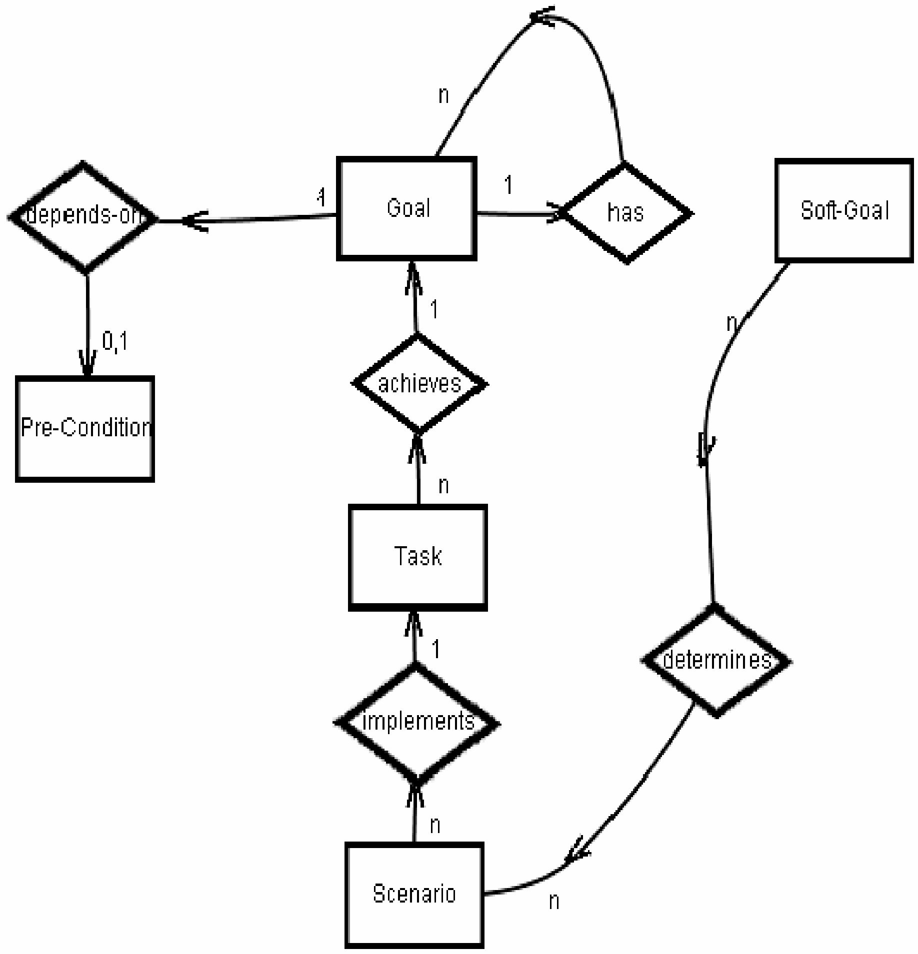
- Vision (elicitation) : explore elicitation techniques to drive goal variability models (may use scenarios as well, as in L' Ecritoire)
- Vision (analysis): use goal analysis to check for consistency and build variants
- Vision (design): variation points occurring at three levels. Possibility of hiding complexity by using variation points, that is factoring or relationships as variation points.
- Vision (implementation): scenarios are implemented either by agents or by composition of state machines (see David's Harel recent work) or traditional OO (frameworks see Fontoura and Lucena).

How

- What we have so far:
  - Initial model (first and latest version)
  - Initial modeling language
  - Initial example based on the Light Control exemplar (http://rn.informatik.uni-kl.de/~recs/)

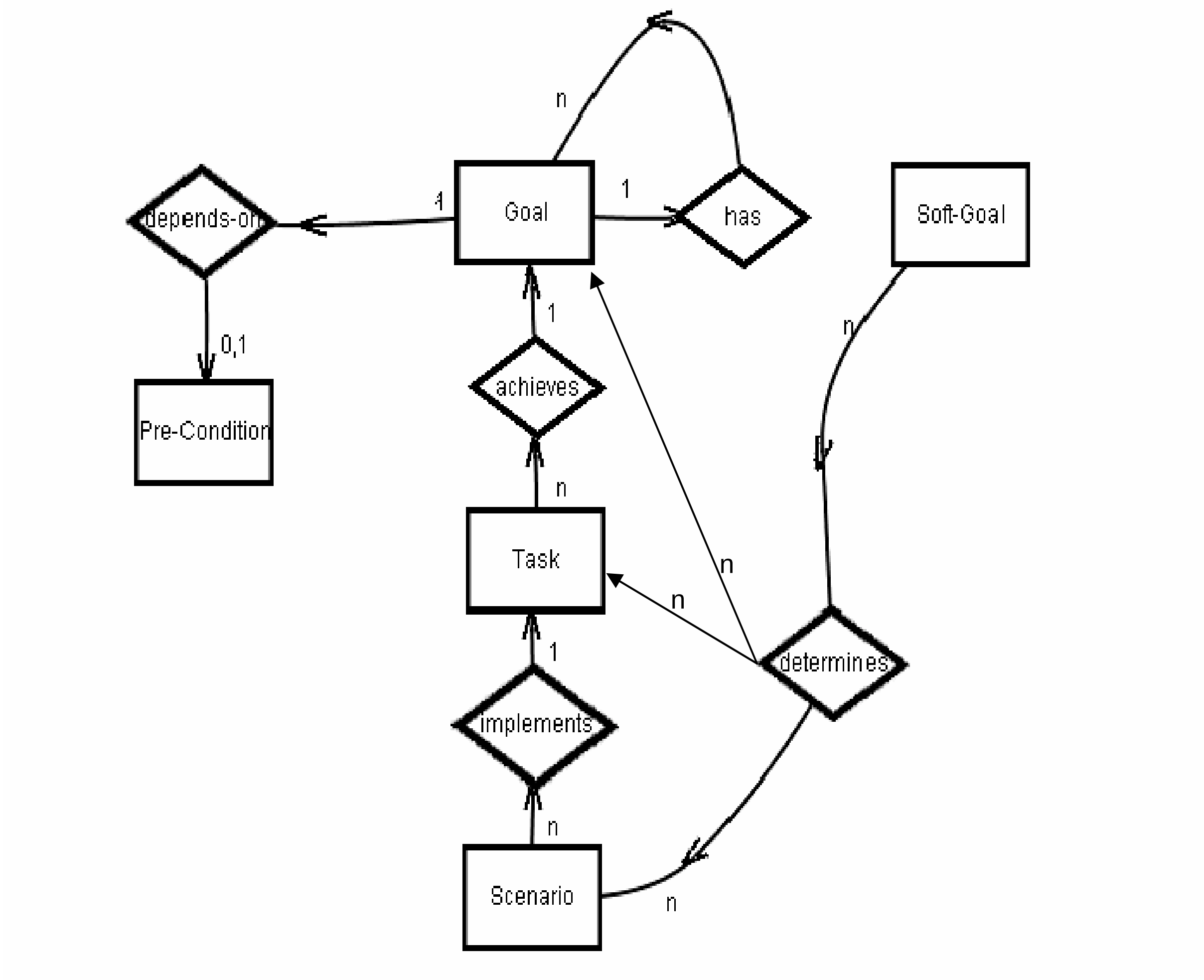
How

Initial Model



How

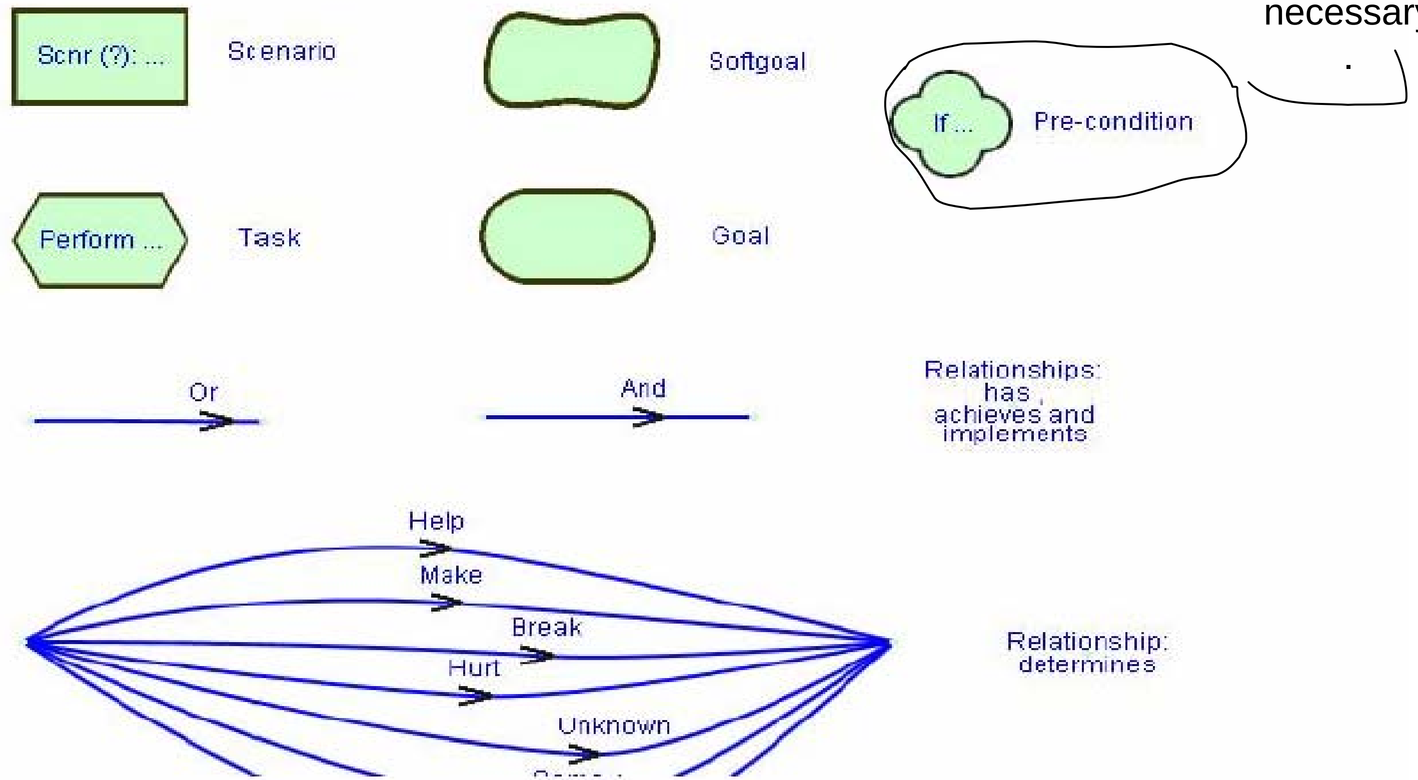
New Model



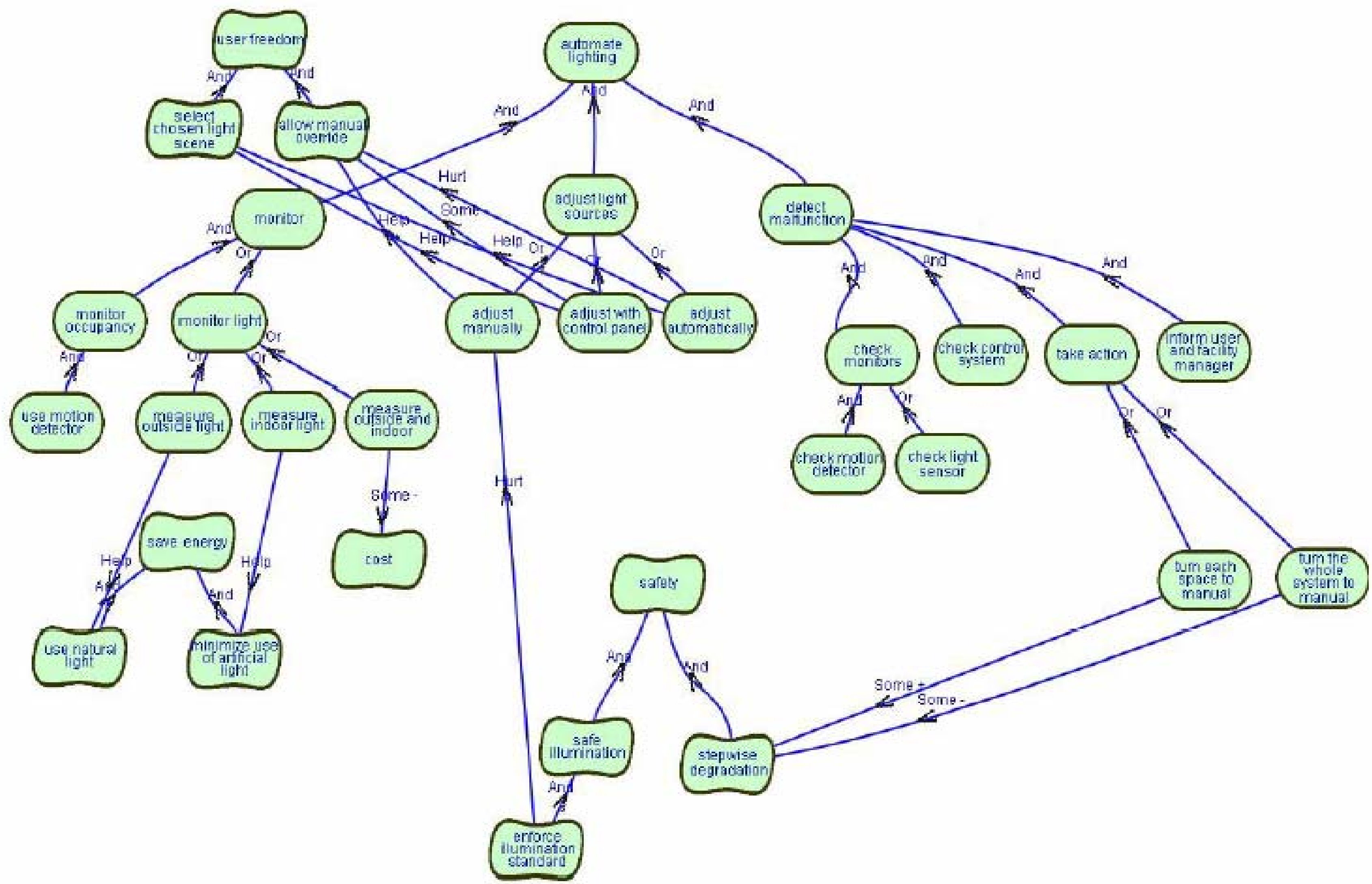
How

Initial Modeling Language

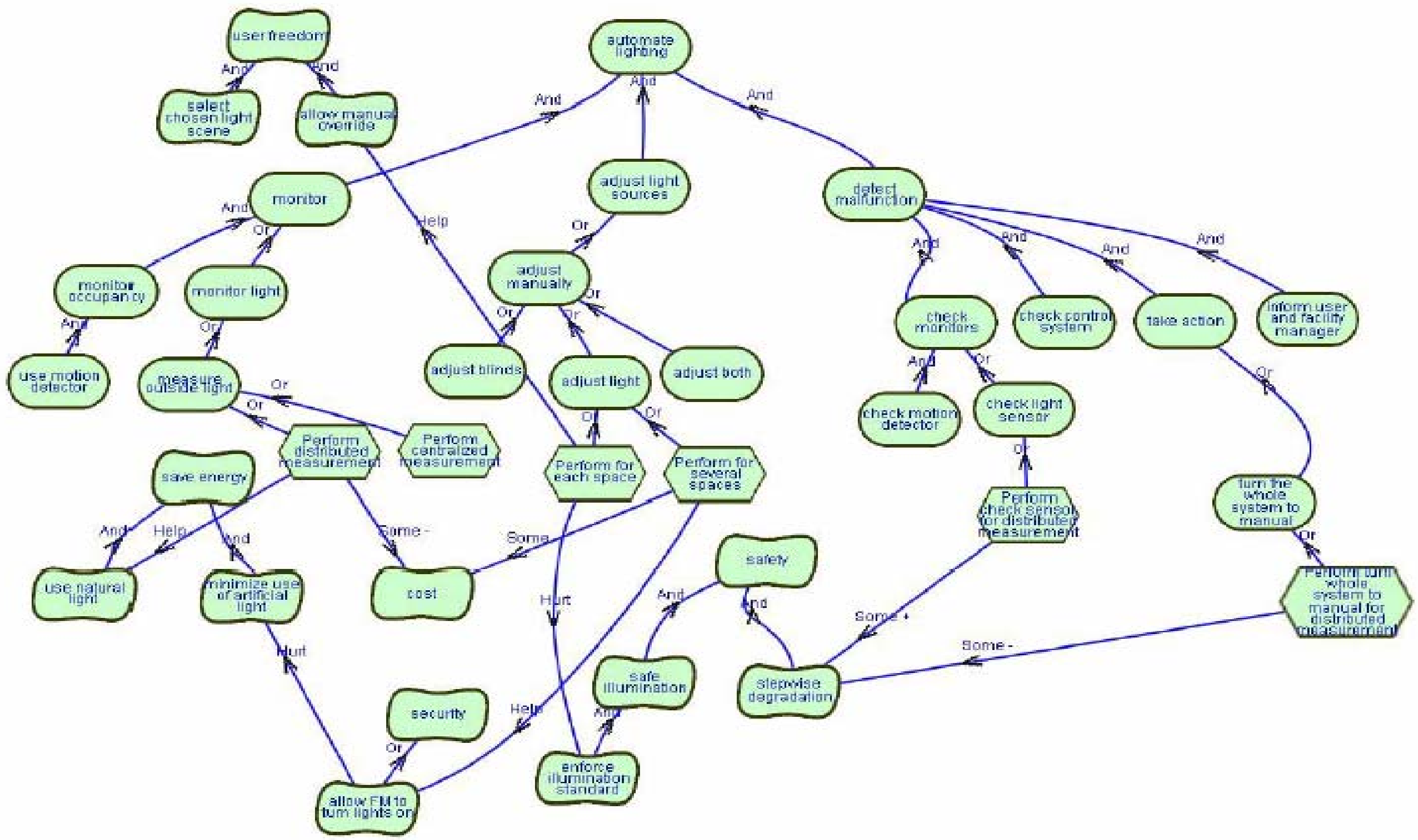
Do not know if it is really necessary



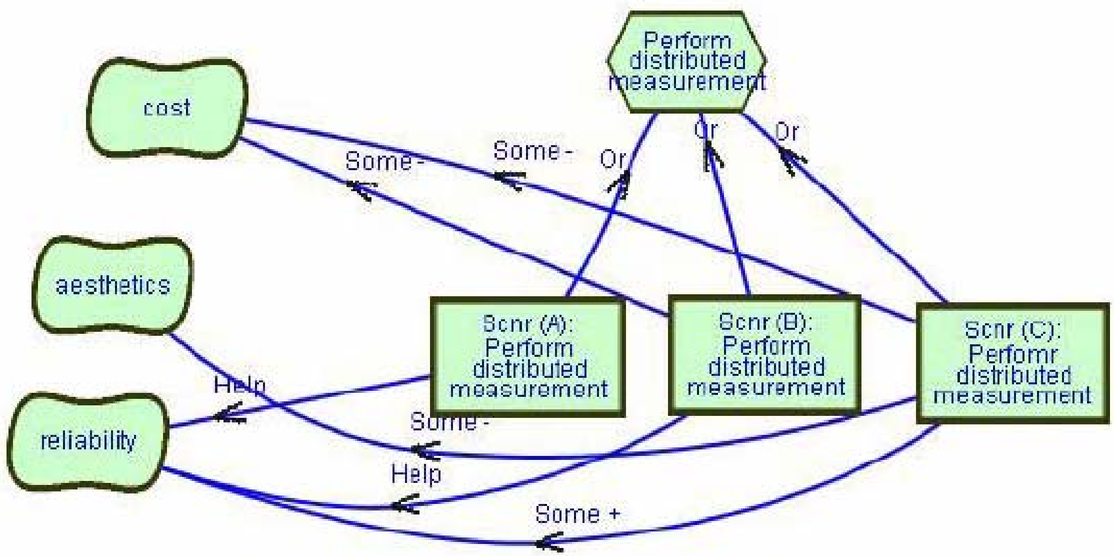
How (1st level of variability)



How (2nd level of variability)



How (3rd level of variability)



How (3rd level of variability)

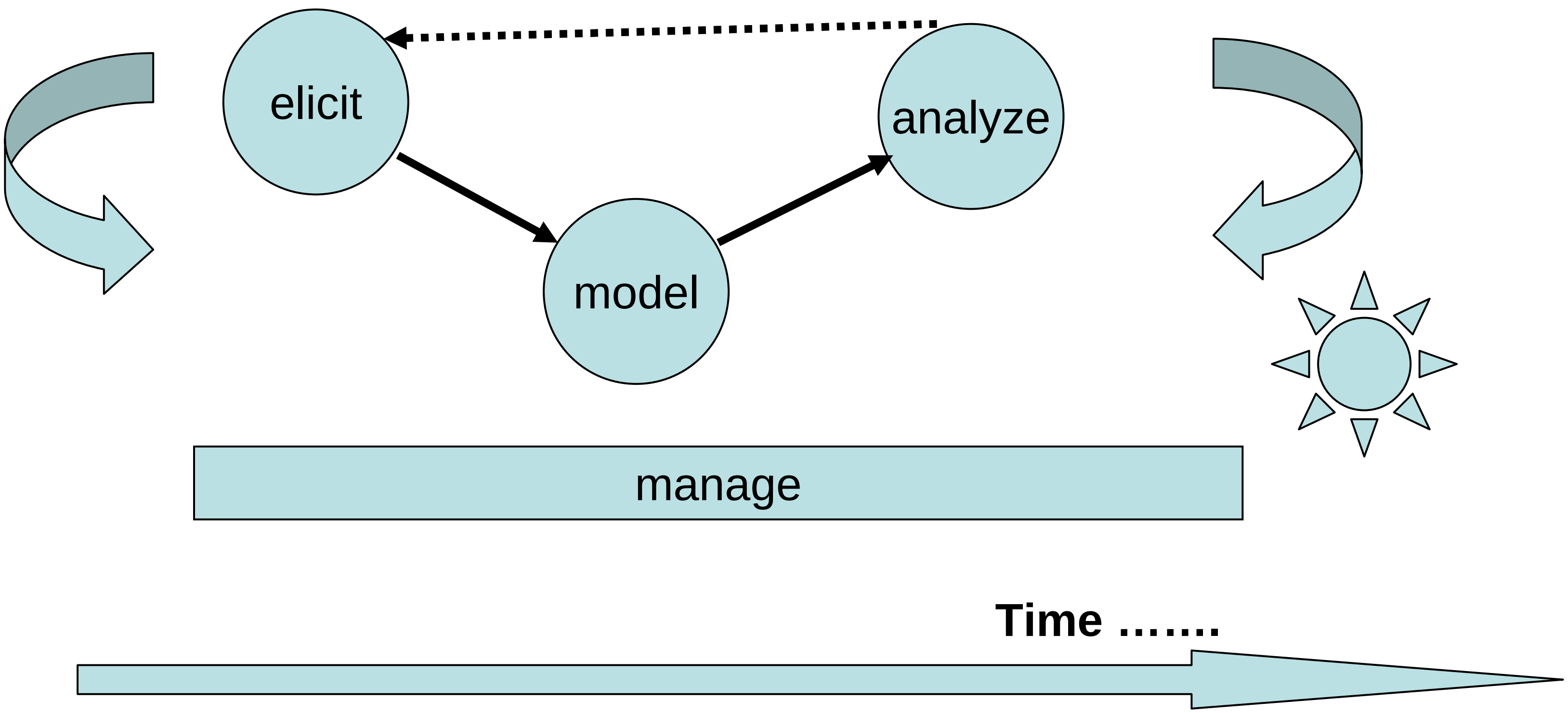
	perform distributed measure (A)	Scenario B	Scenario C
Context:			
temporal location	interval between sunrise and sunset	interval between sunrise and sunset	interval between sunrise and sunset
geographical location	outside fixed unit one window	outside fixed unit one window	outside measured unit two windows
pre-condition	control system ok; designated sensor ok; designated converter ok	control system ok; designated sensor ok; designated converter ok	control system ok; designated sensor ok; designated converter ok
Resources:	sensor	sensor	two sensors
Skills	resolution: 1 lux range: 1-10000 lux reaction time: 10 ms camuflaged: yes converter conversion time: 1s	resolution: 0.5 lux range: 0.5-10000 lux reaction time: 5 ms camuflaged: yes converter conversion time: 0.5s	resolution: 1 lux range: 1-10000 lux reaction time: 10 ms camuflaged: no two converters conversion time: 1s
Actors:	sensor reader lux broadcaster	sensor reader lux broadcaster	sensor reader lux broadcaster
Skills	sampling time: T<= (e-1)/R-1	sampling time: T<= (e-1)/R-1	sampling time: (T<= (e-1)/R-1)/2
Preferences	Cost; Aesthetics	Cost; Aesthetics	Cost; Aesthetics; Reliability
Episodes	Sensor reader reads sensor Lux broadcaster transmit lux	Sensor reader reads sensor Lux broadcaster transmit lux	Sensor reader reads sensor 1 Sensor reader reads sensor 2 Lux broadcaster computes compound lux Lux broadcaster transmit lux
Exceptions:	PERFORM MONITORING OF DISTRIBUTED MEASURAMENT (A distributed measurement; Scenario A in perform; Scenario A in perform action for monitoring distributed measurment; Scenario A in perform inform user for monitoring distributed measurment;	Scenario B in perform monitoring of distributed measurement; Scenario B in perform; Scenario B in perform action for monitoring distributed measurment; Scenario B in perform inform user for monitoring distributed measurment;	Scenario C in perform monitoring of distributed measurement; Scenario C in perform; Scenario C in perform action for monitoring distributed measurment; Scenario C in perform inform user for monitoring distributed measurment;

How

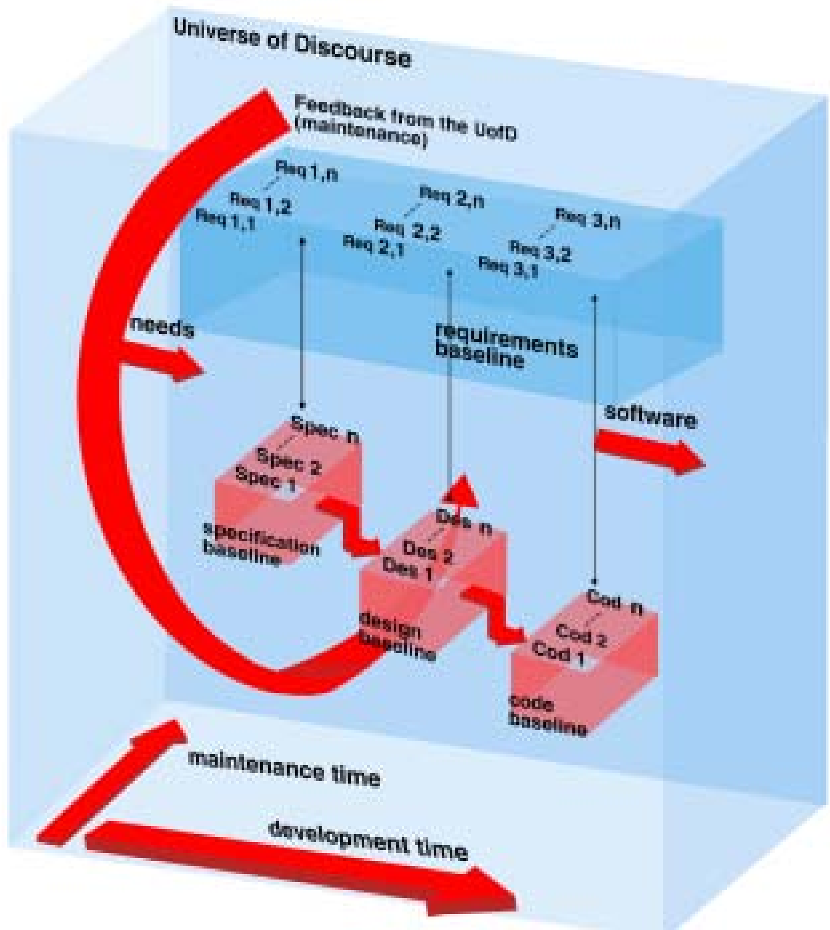
- What we are planning (short term):
  - establish better ways of defining modeling context (like using: objective and viewpoint of the model as in SADT)
  - analysis mechanisms (matrix), similar to the one already being used at UofT-GSP
  - customization selection (as in GSP)
  - variant projection (Dagstuhl model)
  - traceability (rationale)

Constant Concern

Requirements Process



Constant Concern
Requirements Baseline



Challenges

- What is the difference from domain modeling and generative reuse based on DSLs?
- Dealing with evolution (is it a solution?)
- Exploring elicitation from multiple stakeholders, viewpoints to improve elicitation and viewpoints/perspectives to improve modeling and analysis.
- Is it a feasible platform for further implementation?