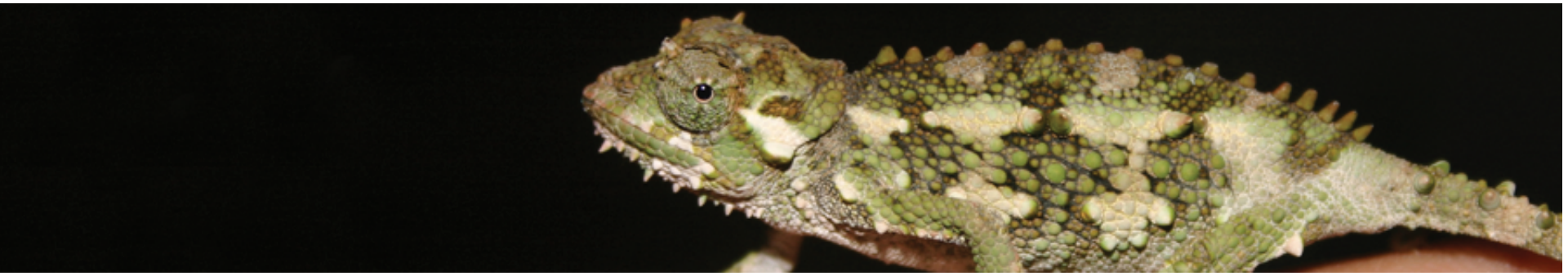


Software Systems Adaptation by Composition

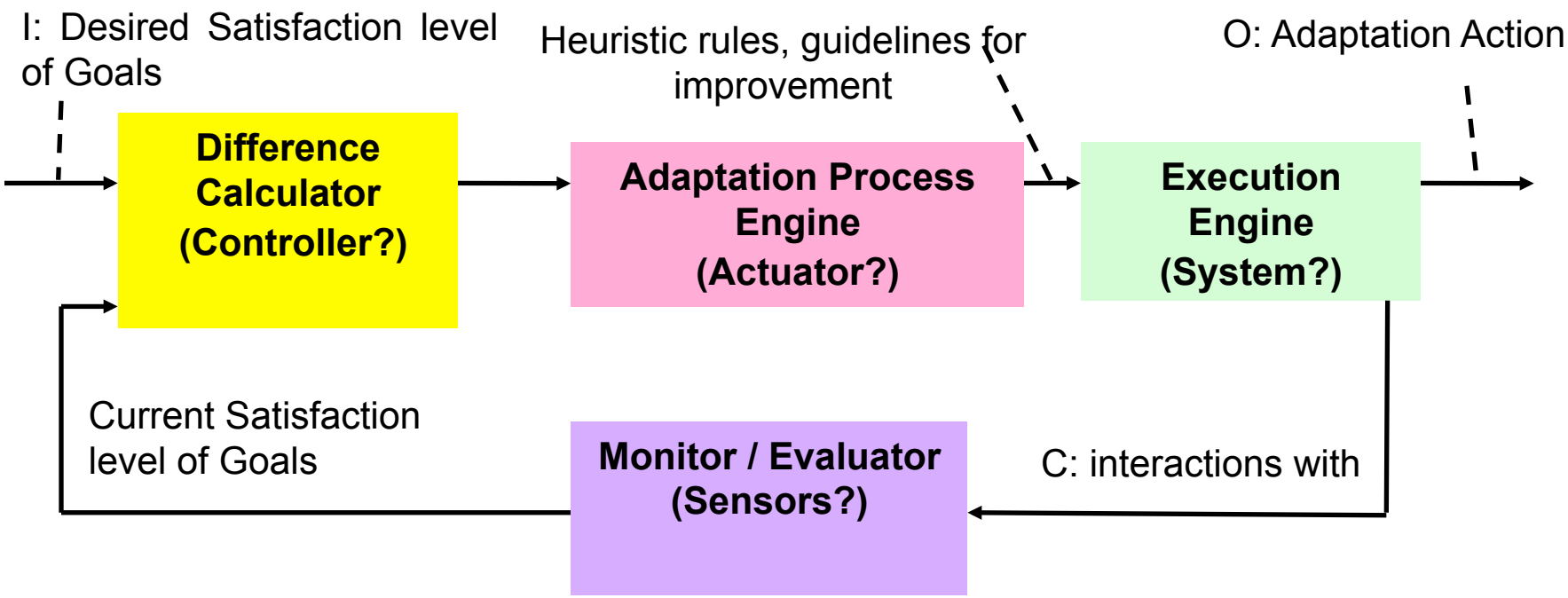
Lin Liu
Information Systems Engineering Group,
School of Software , Tsinghua University, China

Adaptive mechanism in natural world

- Different chameleon species are able to vary their coloration and pattern through specialized cells contain pigments in three layers below their transparent outer skin.
- Color change in chameleons has functions in social signaling and in reactions to temperature and other conditions, as well as in camouflage.
- The relative importance of these functions varies with the circumstances, as well as the species.

Why Engineering Adaptive software?

- **Control the developmental process of software**
- **Control the behavior of software**
- **Control the quality of software**



"Feedback loop" in goal-oriented RE (Liu, et al, IWSC, 2007)

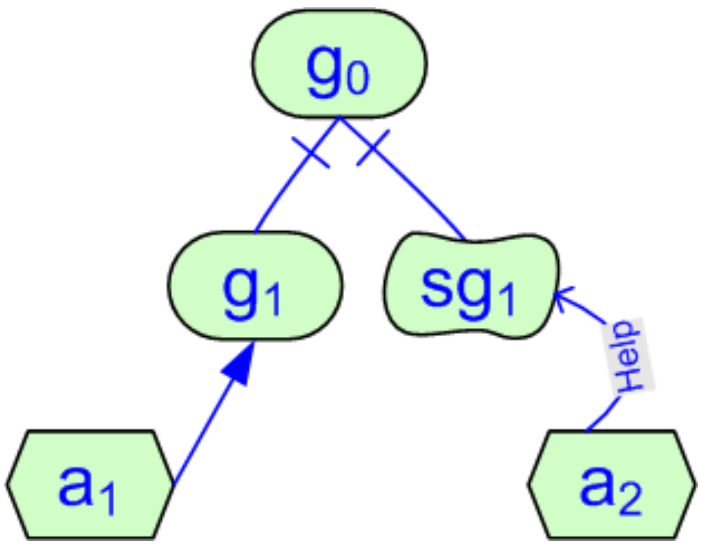
RE@Runtime 2010

A Typology of Systems with Goal Models

with Yunsong Jian, Tong Li, Eric Yu

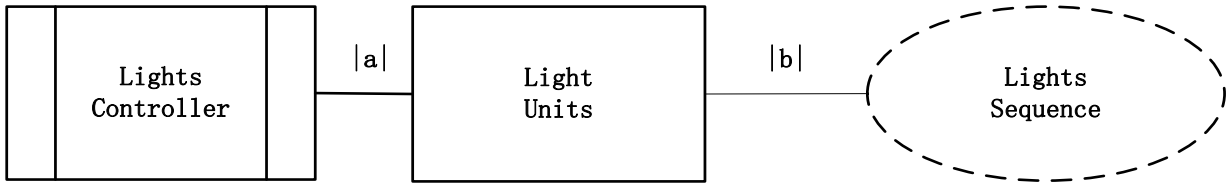
Static Systems Scenario

- the system requirements remain unchanged at run-time.
- It means that the system responds to the operating environment and users' requests in a fixed and predefined way.
- In the requirements model for static systems, the goal set G is fixed, goal refinement and operationalization is determined at design time.
- Environment variables associated to the goal model take constant value.



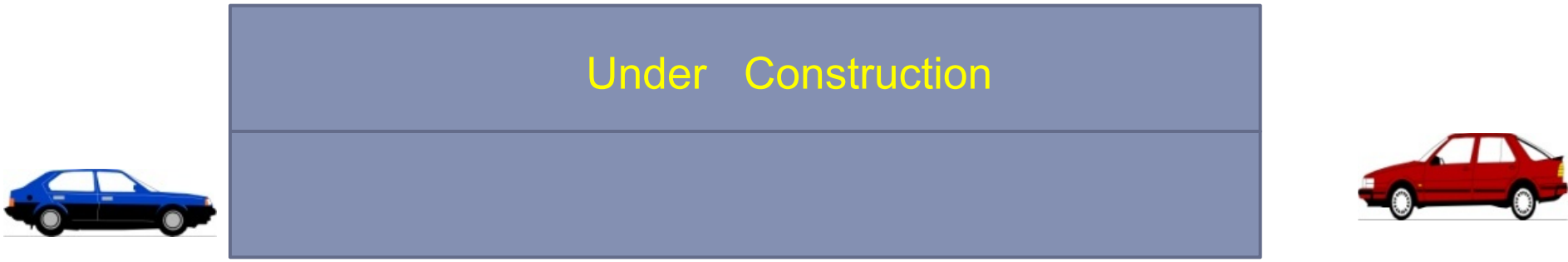
Traffic Light Control Example

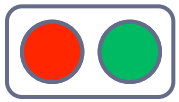
- static




```
|a|::={Rpulse[i],Gpulse[i]}
|b|::={Stop[i],Go[i]}
i::{left,right}
```

(Jackson, 2001)

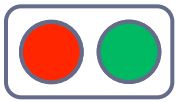




R G

left

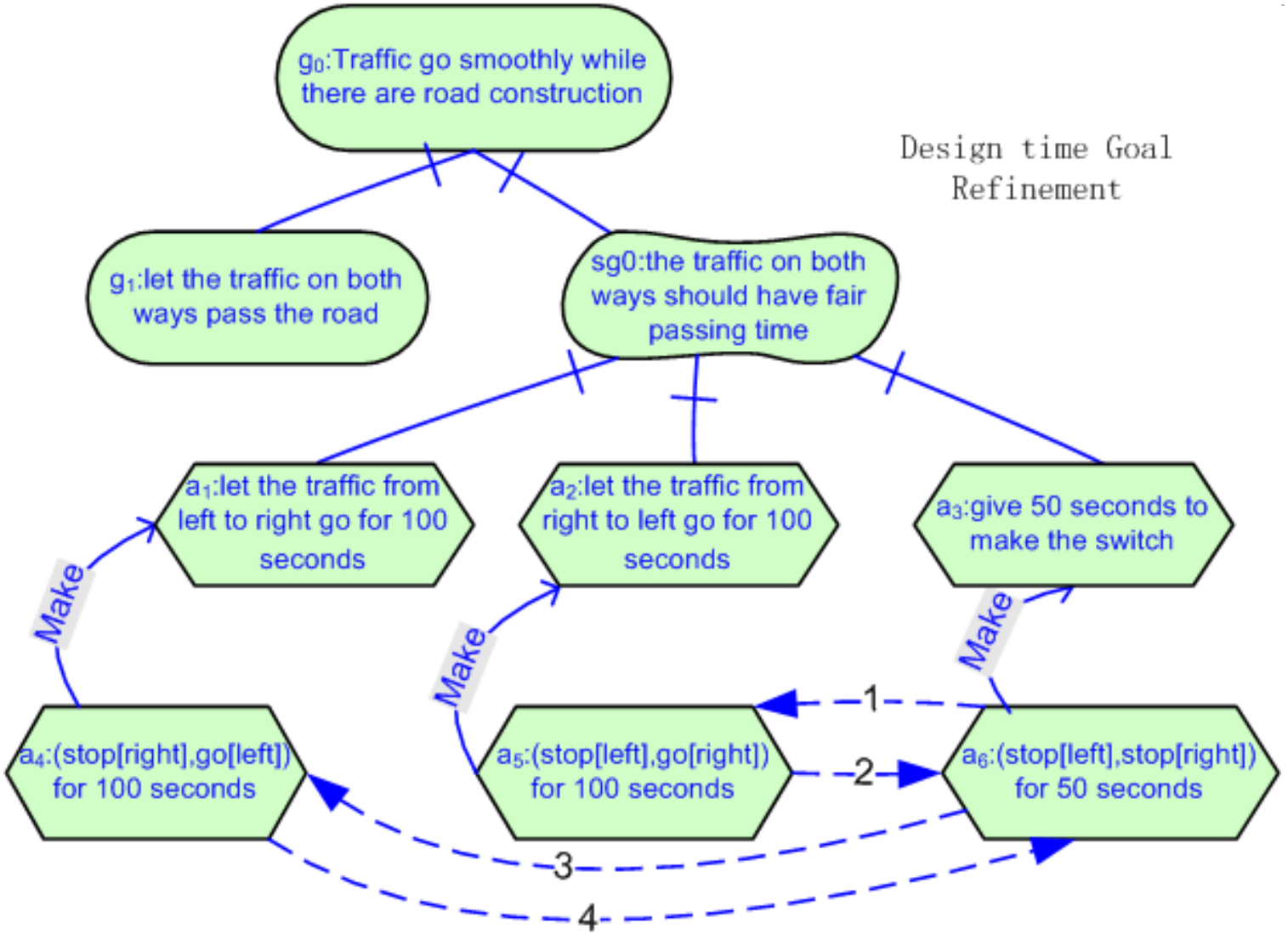
```
(Stop[left],Stop[right]) for 50 seconds;
(Stop[left],Go[right]) for 100 seconds;
(Stop[left],Stop[right]) for 50 seconds;
(Go[left],Stop[right]) for 100 seconds
```



R G

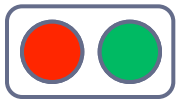
right

Goal Model for the static Traffic Light System



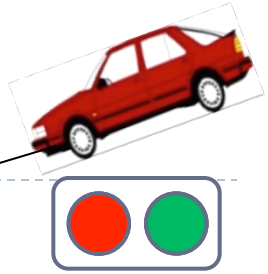
Reconfigured Traffic light

Under Construction



R G

Left

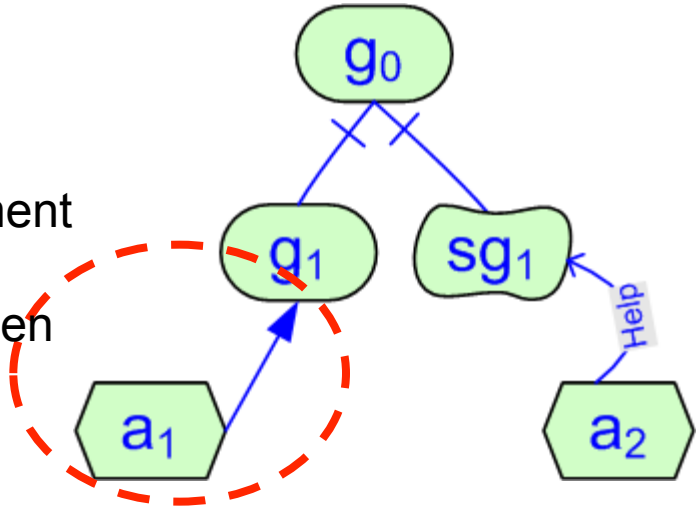


R G

Right

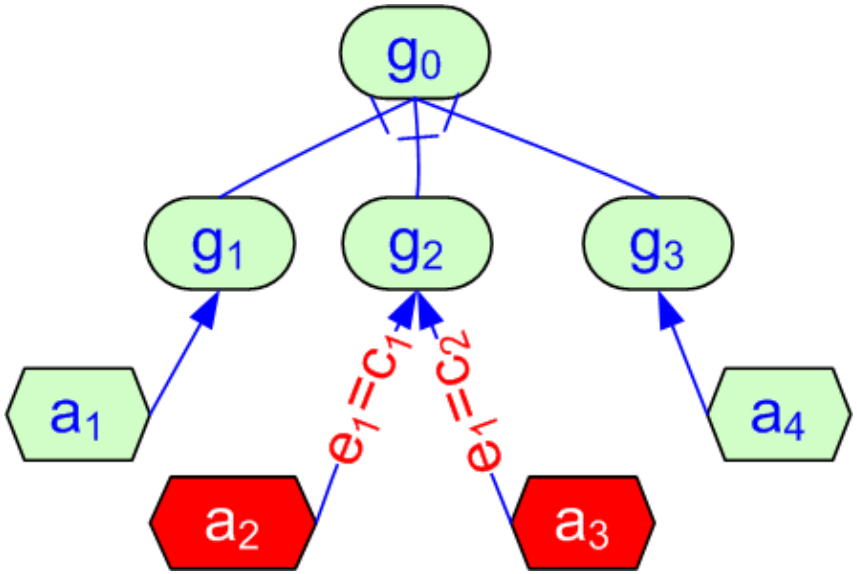
Stop[left],Stop[right] for 50 seconds;
Stop[left],Go[right] for 75seconds;
Stop[left],Stop[right] for 50 seconds;
Go[left],Stop[right] for 150 seconds;

Observe the environment at deployment time, change the setting when necessary

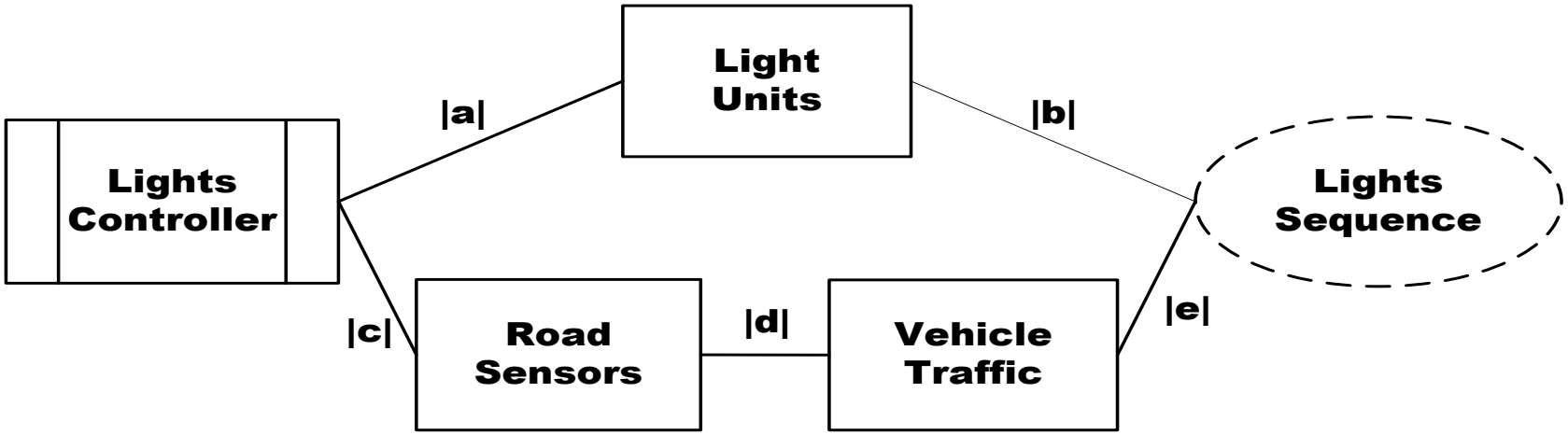


The Sensitive/Reactive Systems Scenario

- the system does change its behavior at run-time and the system could respond to certain predefined operating environmental changes and users' requests.
- This kind of systems has some adaptation ability but the ability is limited by a set of predefined rules.
- The adaptation is the result of non-deterministic goal refinements at design time.
- The achievement of goals depends on the states of the run-time environmental variables. This is a vivid example of gradually adding uncertainty into requirements analysis and system development.



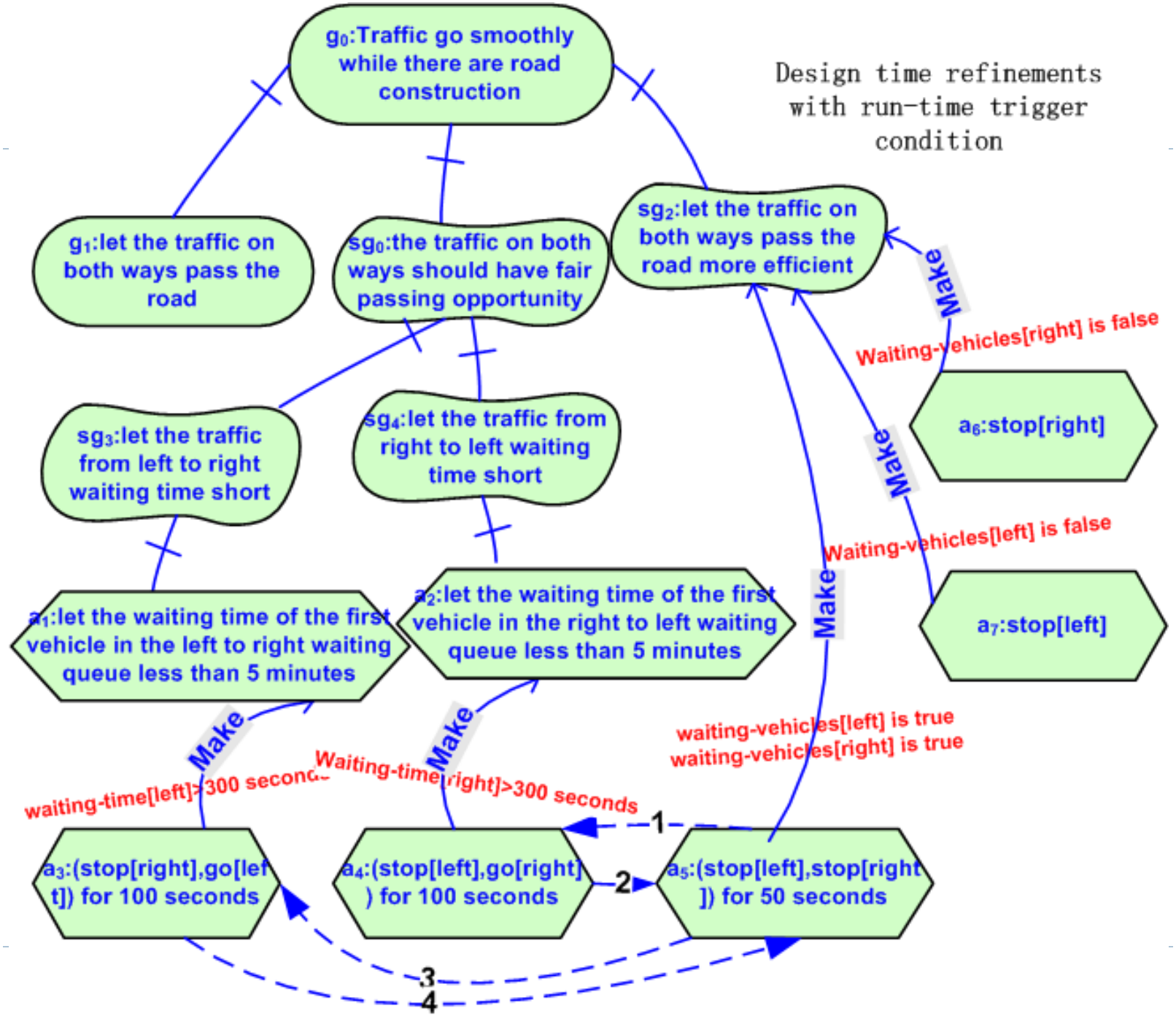
Sensitive One-way Traffic Light System



|a|::={Rpulse[i],Gpulse[i]}
|d|::={VehiclePassSensor[i]}
|b|::={Stop[i],Go[i]}
|c|::={SensorOn[i]}
|e|::={VehicleTraversing[v]}
i::{left,right}

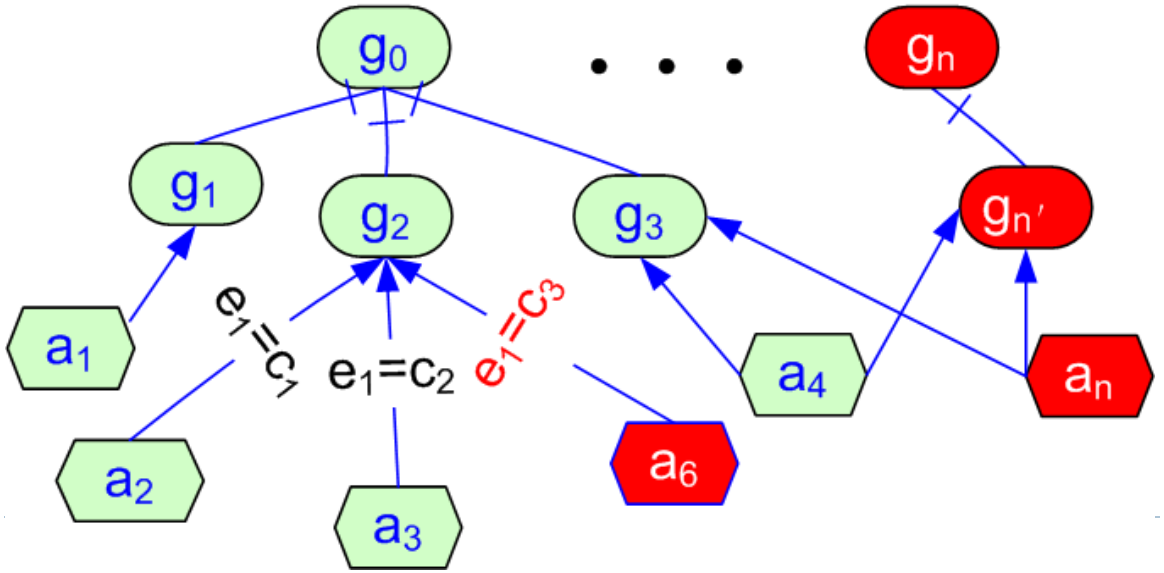
Under Construction

Design time refinements with run-time trigger condition



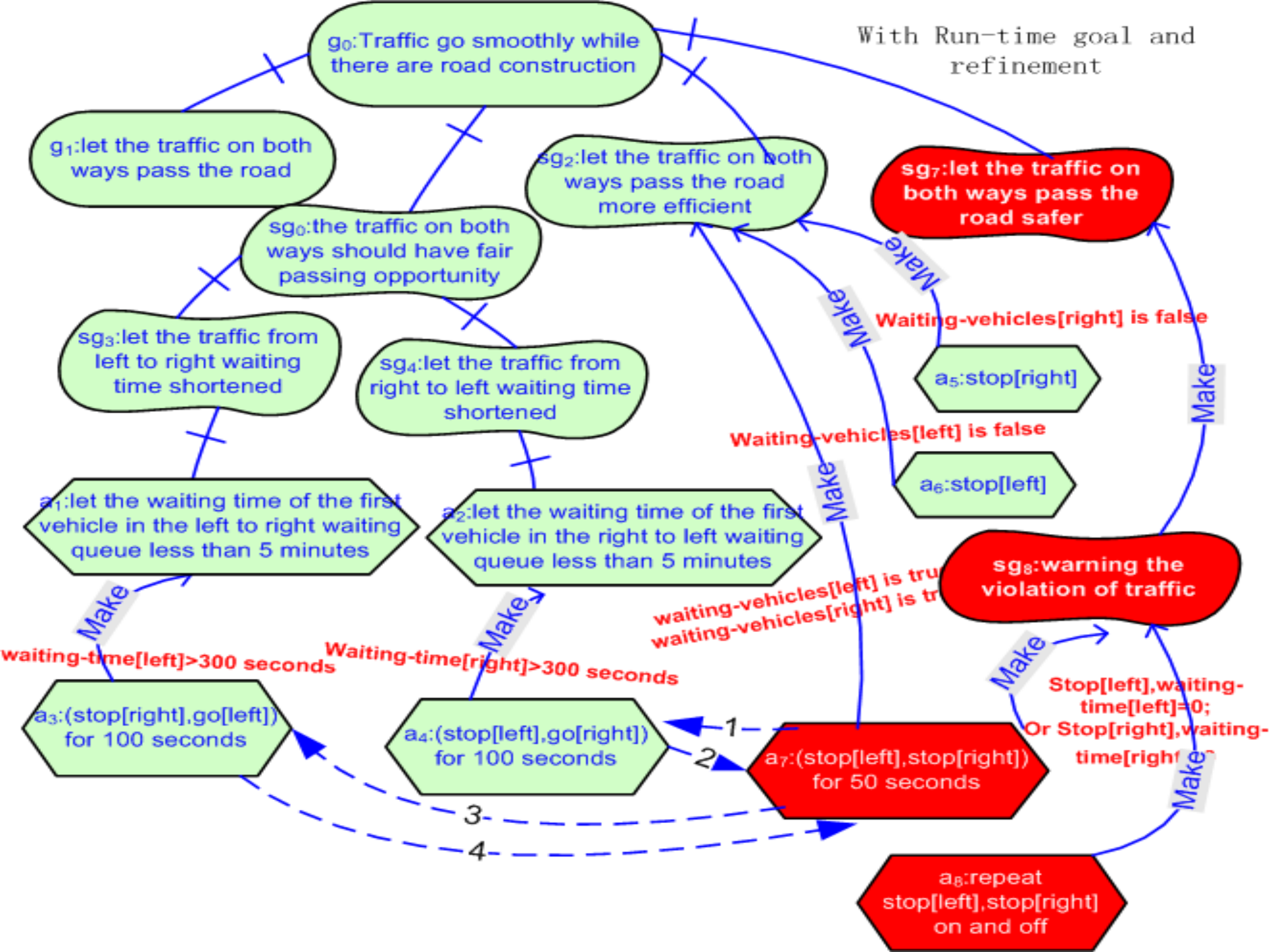
The Run-Time Adaptation Scenario

- the system requirements changes and the system adaptation actions and the adaptation rules are not completely predefined, there is greater uncertainty at the design time.
- At first the adaptation rules set R only contain some predefined rules.
- When the system is running, in order to handle the changes of the operational environment and the users' needs, new goals, new environment conditions, new activities and new rules are added to the sets G, E, A and R at run-time respectively.



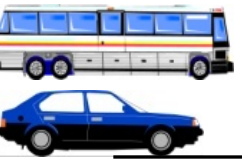
Adaptive One-Way Traffic Light System

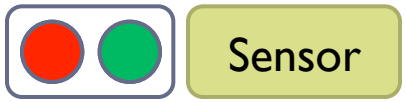
Under Construction


With Run-time goal and refinement
g0:Traffic go smoothly while there are road construction
g1:let the traffic on both ways pass the road
sg0:the traffic on both ways should have fair passing opportunity
sg2:let the traffic on both ways pass the road more efficient
sg7:let the traffic on both ways pass the road safer
sg3:let the traffic from left to right waiting time shortened
sg4:let the traffic from right to left waiting time shortened
a1:let the waiting time of the first vehicle in the left to right waiting queue less than 5 minutes
a2:let the waiting time of the first vehicle in the right to left waiting queue less than 5 minutes
a5:stop[right]
a6:stop[left]
sg8:warning the violation of traffic
Waiting-vehicles[right] is false
Waiting-vehicles[left] is false
waiting-vehicles[left] is tru
waiting-vehicles[right] is tr
waiting-time[left]>300 seconds
Waiting-time[right]>300 seconds
Stop[left],waiting-time[left]=0;
Or Stop[right],waiting-time[right
a3:(stop[right],go[left]) for 100 seconds
a4:(stop[left],go[right]) for 100 seconds
a7:(stop[left],stop[right]) for 50 seconds
a8:repeat stop[left],stop[right] on and off
Make
1
2
3
4

Collaborative One-way Traffic Light System

Under Construction





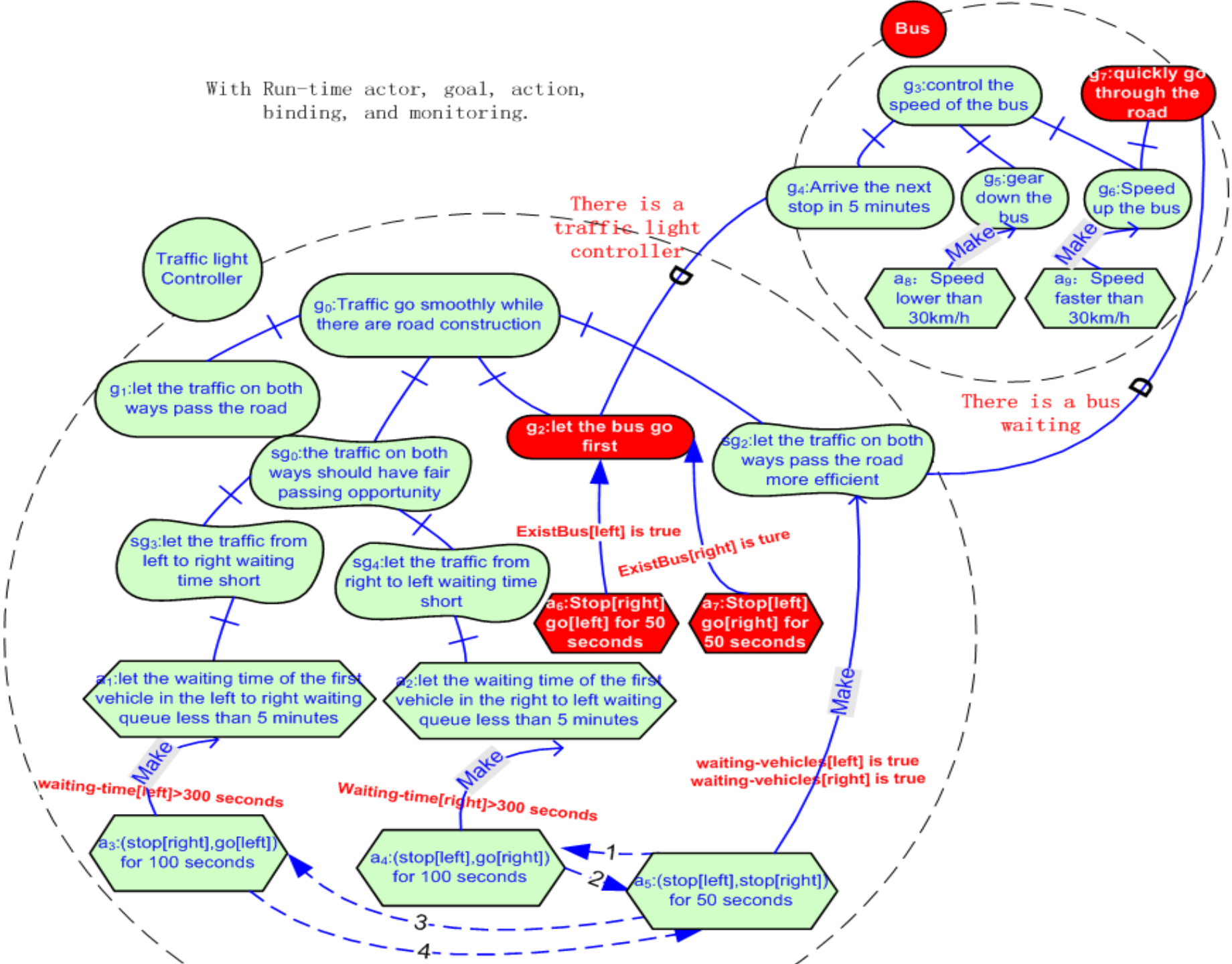
R G

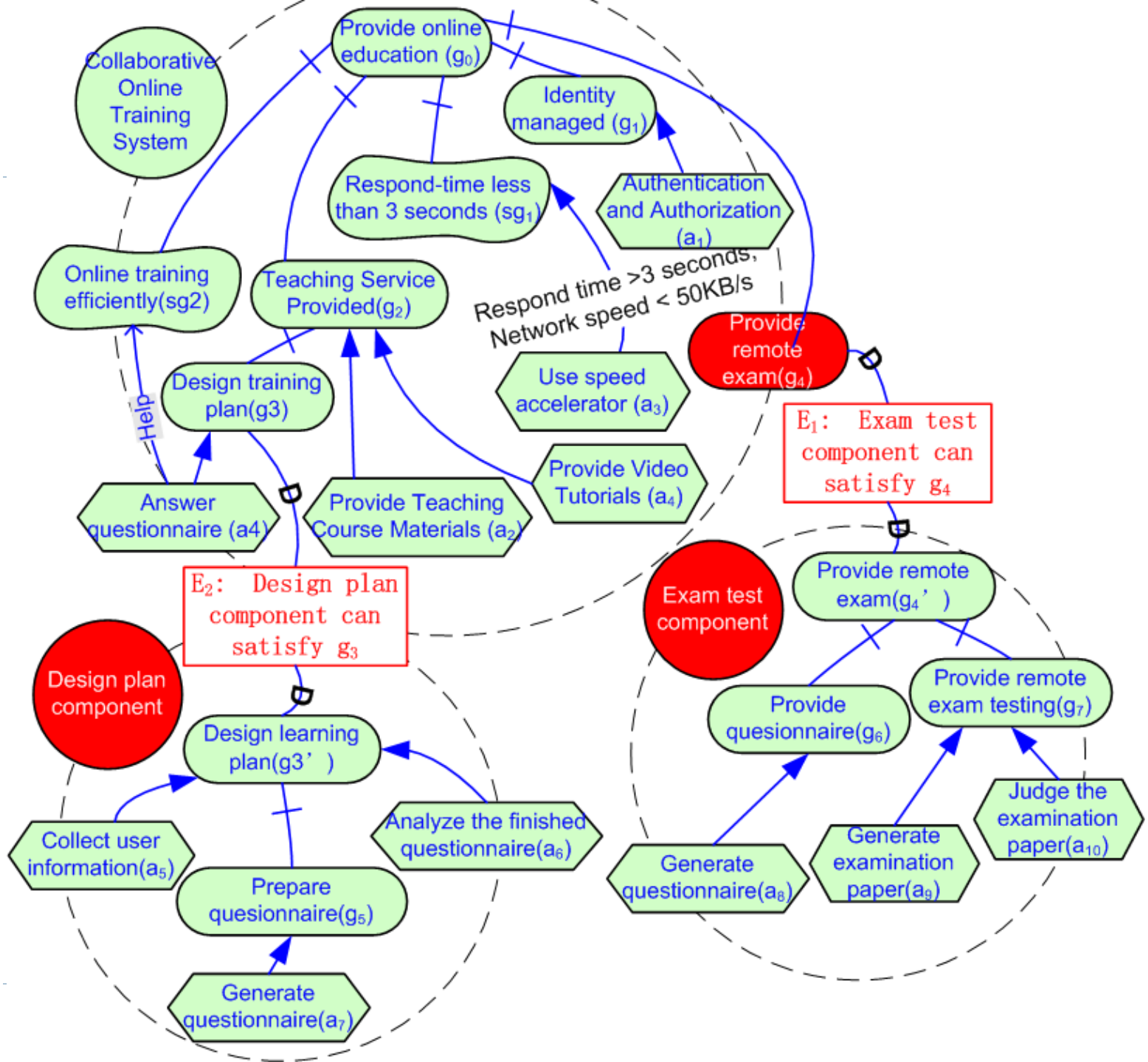
left

R G

right

With Run-time actor, goal, action, binding, and monitoring.




Collaborative Online Training System
Provide online education (g_0)
Identity managed (g_1)
Respond-time less than 3 seconds (sg_1)
Authentication and Authorization (a_1)
Online training efficiently(sg2)
Teaching Service Provided(g_2)
Respond time >3 seconds, Network speed < 50KB/s
Provide remote exam(g_4)
Use speed accelerator (a_3)
Design training plan(g3)
Help
E_1: Exam test component can satisfy g_4
Provide Video Tutorials (a_4)
Answer questionnaire (a4)
Provide Teaching Course Materials (a_2)
E_2: Design plan component can satisfy g_3
Exam test component
Provide remote exam(g_4')
Design plan component
Provide remote exam testing(g_7)
Provide quesionnaire(g_6)
Design learning plan(g3')
Collect user information(a_5)
Analyze the finished questionnaire(a_6)
Judge the examination paper(a_{10})
Generate examination paper(a_9)
Generate questionnaire(a_8)
Prepare quesionnaire(g_5)
Generate questionnaire(a_7)

Collaboration

- The software collaboration means that the software entities are being aware of the existence of others, and could collaborate to achieve a common goal.
- There are delegation and informing activities between them.
- The states of other components are partially observable by other components, as environment variables.
- An abstract collaboration model for components has associated goal-action refinement structures, interconnected, and maintained through protocols supporting the requesting, publication, searching, binding, delegation and revoking relationships among individual components.

Adaptive Design in Response to Changing Requirements

- Systems as organisations
  - Framework that facilitates the development of adaptive **role**-based system structure, with role-role **contracts**
- Contracts define (rich) interaction protocols and measure performance – implemented with association aspects
- Strict separation of role from player – roles define the function, players do the work: role-player bindings
- A role is stateful and is defined as a set of provided/ required service interfaces
- Organisation (system) is a recursive hierarchy of self-managed composites, with each having an organiser role
- Roles can be filled by players of different capabilities
  - Services, Objects, Self-managed composites, Agents, Humans

Summary

- Composition-based development of software systems from a set of software components distributed over the Internet.
  - ideally, the software components, are autonomous, self-contained and distributed.
  - It is able to respond to the perceived changes in the environment by means of reconfiguration and reorganization
  - It collaborates with one another on demand, which adopts an iterative composition process turning various "disordered" resources into an "ordered" software system.
- brings promptness, flexibility and reusability in building distributed software systems.
- As a new requirement emerges, or a new software technology becomes available, or an unexpected change happens, existing requirements-solutions bindings can be adjusted for optimization purpose.

An Analogue: Software vs. Data

- Software are like the pipes and pumps and storage tanks in a plumbing system
- Data is like the water flowing through those pipes
- And there is also data about the requirements, design and run-time status of the software…
- If you suspected that your water was poisoned, Would you call the plumber ? What if you saw water leaking, or reduction of water volume, …?

Call for Papers and Panel Proposals

11th International Workshop on Software Cybernetics

IWSC 2014

In conjunction with COMPSAC 2014, July 21-25, 2014

Vasteras, Sweden, July 21, 2014

Thanks!