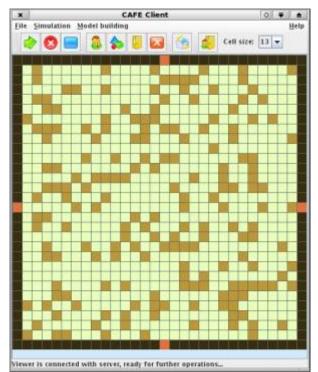
Open Smart Meter

Actually this is a project proposal. The idea is to establish a project of open smart meter based on open design and open software principles. Everyone could participate in this project. The project management has been entrusted to the project board.

See the presentation (in Polish): Open Smart Meter - Idea

Cafe

Cafe - which means: Cellular Automata Framework Environment. The aim of this project is to create a Java based CA modeling tool which allows to create and simulate various kinds of asynchronous, nonhomogenous cellular automata. The automata built this way will be used to simulation of pedestrians behaviors (*MNiSW grant: N N516 228735*), and to modeling robots behavior. In the robotic approach CA is treated as another kind of knowledge used by so-called *Intelligent Control System*. Screen of the Cafe's early version can be viewed below.



file dokuwiki_fancy_my.tpl does not exist

RAT

Proposed approach to executable modeling of ractive systems consist of two elements: UML editor and Reactive Appliance Toolkit. There is no limitation to one specific kind of UML editor. Every editor could be used if satisfy following requirements: at least metamodel 1.4, XMI support, stereotypes and profiles, class diagram, statechart diagram, activity diagram. In order to use Reactive Appliance Toolkit standard Java 1 and some implementation of RealTime Java (e.g. Jamaica VM 2) should be instaled. Reactive Appliance Toolkit consist of three components:

- API library
- Code generator
- Management console

A console for the robot control case study is shown below:

	consis		080
🗇 results.Rabel	o' a' 🖸		and the second second
Applance details	Events list		Sijava CRescu
Applance name (results Robot	gota		1. 4 8
Actual state results Robot CheriMMiskerGensor2	changeDirector	-	Line we we
Applaine Ho	stopRobet WaveForward		
iutikas 🔊	atepDeck	Load applance Make cascad	31)
velocity=5.	100000		
Materiale,		Console	
Whidee-true;	partnerse .	Event results Protection/Protect vol12	
	Fite	Event results Robultine geb rectar	
		4 P	
() and we have a set of the bound of the object of the action of the set of t	2017 23:03 890		
	101 - 12 00 80-0		
cess message *:getappliances		bot AnonTransitional Make Ste	
cess message *:getappliances 0: [07:24:27][Robot] Found automatic tra	nsition [Stop.Rol		pBack.] for state [Stop]
cess message *:getappliances D: [07:24:27][Robot] Found automatic tra D: [07:24:27][Robot] appliance [results.F	nsition [Stop.Rol		pBack.] for state [Stop]
cess message *:getappliances Dr. [07:24:27][Robot] Found automatic tra Dr. [07:24:27][Robot] appliance [results.F cess message *:getappliances	nsition [Stop.Rol lobat] moves fro	m [results.Robot.Stop] to [Ma	pBack.] for state [Stop] keStepBack] automatically v
cess message *:getappliances D: [07:24:27][Robot] Found automatic tra D: [07:24:27][Robot] appliance [results.F cess message *:getappliances D: [07:24:27][Robot] Found automatic tra	nsition (Stop.Rol tobot) moves fro nsition (MakeSto	m [results.Robot.Stop] to [Ma pBack.Robot AnonTransition]	pBack.] for state [Stop] keStepBack] automaticaly v 5.CheckWhiskerSensor2.] fo
dere agestatet reporter postpret op soo cess message *:getappliances 0. (07:24:27)[Robot] appliance [results.5 cess message *:getappliances 0. (07:24:27)[Robot] Found automatic tra 0. (07:24:27)[Robot] Found automatic tra	nsition (Stop.Rol tobot) moves fro nsition (MakeSto	m [results.Robot.Stop] to [Ma pBack.Robot AnonTransition]	pBack.] for state [Stop] keStepBack] automaticaly v 5.CheckWhiskerSensor2.] fo
cess message *:getappliances 0:(07:24:27)[Robot] Found automatic tra 0:(07:24:27)[Robot] appliance [results.F cess message *:getappliances 0:(07:24:27)[Robot] Found automatic tra 0:(07:24:27)[Robot] appliance [results.F	nsition (Stop.Rol tobot) moves fro nsition (MakeSto tobot) moves fro	m [results.Robot.Stop] to [Ma pBack.Robot_AnonTransition! m [results.Robot.MakeStepBa	pBack.] for state [Stop] keStepBack] automaticaly v 5.CheckWhiskerSensor2.] fo
cess message * .getappliances b. (07:24:27)[Robot] Found automatic tra b. (07:24:27)[Robot] appliance [results.5 cess message * .getappliances b. (07:24:27)[Robot] cound automatic tra b. (07:24:27)[Robot] appliance [results.5 (07:24:27)[Robot] appliance [results.5 (07:24:27)[Robot] appliance [results.5]	nsition (Stop. Rol tobot) moves fro nsition (MakeSto tobot) moves fro ling dispatcher f	m [results.Robot.Stop] to [Ma pBack.Robot_AnonTransition! m [results.Robot.MakeStepBa	pBack.] for state [Stop] keStepBack] automaticaly v 5.CheckWhiskerSensor2.] fo
cess message * .getappliances b. (07:24:27)[Robot] Found automatic tra b. (07:24:27)[Robot] appliance [results.5 cess message * .getappliances b. (07:24:27)[Robot] cound automatic tra b. (07:24:27)[Robot] appliance [results.5 (07:24:27)[Robot] appliance [results.5 (07:24:27)[Robot] appliance [results.5]	nsition (Stop. Rol tobot) moves fro nsition (MakeSto tobot) moves fro ling dispatcher f	m [results.Robot.Stop] to [Ma pBack.Robot_AnonTransition! m [results.Robot.MakeStepBa	pBack.] for state [Stop] keStepBack] automaticaly v 5.CheckWhiskerSensor2.] fo
cess message * :getappliances 0: (07:24:27)[Robot] Found automatic tra- 0: (07:24:27)[Robot] appliance [results.5 cess message * :getappliances 0: (07:24:27)[Robot] Found automatic tra 0: (07:24:27)[Robot] appliance [results.5 0: (07:24:27)[CheckWhitskerSensor2] Call 0: (07:24:27)[CheckWhitskerSensor2] Call 0: (07:24:27)[CheckWhitskerSensor2] Call	nsition (Stop. Rol tobot) moves fro nsition (MakeSto tobot) moves fro ing dispatcher fi ent: stepBack	m [results.Robot.Stop] to [Ma pBack.Robot_AnonTransition! m [results.Robot.MakeStepBa	pBack.] for state [Stop] keStepBack] automaticaly v 5.CheckWhiskerSensor2.] fo
sess message *:getappliances 2: (07:24:27)[Robot] Found automatic tra- 0: (07:24:27)[Robot] poliance [results.5: cess message *:getappliances 2: (07:24:27)[Robot] Found automatic tra- 2: (07:24:27)[Robot] appliance [results.3: 2: (07:24:27)[CheckWhiskerSensor2] Call 2: (07:24:27)[CheckWhiskerSensor2] Call 3: (07:24:27)[CheckWhiskerSensor2] Call 4: (07:24:27)[CheckWhiskerSensor2] Call 4: (07:24:27)[CheckWhiskerSensor2] Call 4: (07:24:27)[CheckWhiskerSensor2] Call 4: (07:24:27)[CheckWhiskerSensor2] Call 5: (07:24	nsition [Stop.Rol lobot] moves fro nsition [MakeSts lobot] moves fro ing dispatcher f ent: stepBack	m [results.Robot.Stop] to [Ma pBack.Robot.AnonTransition] m (results.Robot.MakeStepBa or raise event [stepBack]	pBack.] for state [Stop] keStepBack] automaticaly v 5.CheckWhiskerSensor2.] fo ck] to [CheckWhiskerSensor

file dokuwiki fancy my.tpl does not exist

Hekate

HeKatE means Hybrid Knowledge Engineering. It is the research project regards Software Engineering based on Knowledge Engineering. The scope of the project includes but it is not limited to Knowledge Representation, Processing and Visualization methods. More information is available on project web page

file dokuwiki_fancy_my.tpl does not exist

