Master's Thesis

The Symbolic Modeling of Complex Functions with Usage of the Enlarging Technique

Student : Kotyk A.I, M.S Informatics

Advisor: PhD O. Letichevskij, senior fellow in Glushkov Institute of Cybernetics of NAS Ukraine

Co-advisor: Dipl.-Ing. Dr. W. Schreiner, A.Univ.Prof.in Johannes Kepler University of Linz, Austria

Overview

1. Previous research. GARP
2. BPSL
3. Example of the project
4. Industrial Application
5. Formulation of the problem
6. Future program
7. References

Previous research. GARP

The specifications(IEEE 802.1D-2004) contained the informal description of the model and 2 state machines describing the transition between states upon receiving certain messages.

GARP Architecture

- GARP Participant: Application + GID
- Each port (on a bridge or a workstation) has a connected Participant
- Bridge participants are connected by GARP Information Propagation (GIP)
- GARP Information Directory (GID) has FSMs for each attribute:
  - Applicant State Machine
  - Registrar State Machine (optional)
- GID has also the only FSM:
  - LeaveAll State Machine (optional)

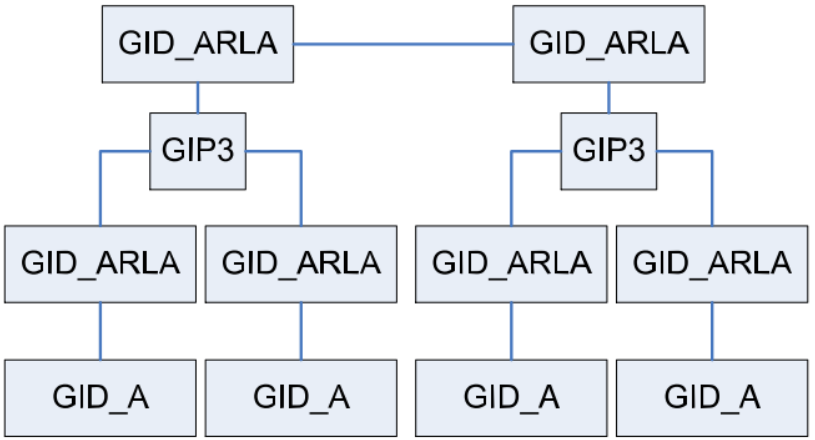
STATE \ EVENT	leavetimer!	transmitPDU!	rJoinIn	rJoinEmpty	rEmpty	rLeaveIn	rLeaveEmpty, LeaveAll	ReqJoin	ReqLeave	Initialize
VA.MT	-x-	AA.MT	AA.IN	VA.IN	VA.MT	VA.MT	VP.MT	-x-	LA.MT	VO.MT
VA.LV	VA.MT	AA.LV	AA.IN	VA.IN	VA.LV	VA.LV	VP.LV	-x-	LA.LV	VO.MT
VA.IN	-x-	AA.IN	AA.IN	VA.IN	VA.IN	VA.LV	VP.LV	-x-	LA.IN	VO.MT
AA.MT	-x-	QA.MT	QA.IN	VA.IN	VA.MT	VA.MT	VP.MT	-x-	LA.MT	VO.MT
AA.LV	AA.MT	QA.LV	QA.IN	VA.IN	VA.LV	VA.LV	VP.LV	-x-	LA.LV	VO.MT
AA.IN	-x-	QA.IN	QA.IN	VA.IN	VA.IN	VA.LV	VP.LV	-x-	LA.IN	VO.MT
QA.MT	-x-	—	QA.IN	VA.IN	VA.MT	VP.MT	VP.MT	-x-	LA.MT	VO.MT
QA.LV	QA.MT	—	QA.IN	VA.IN	VA.LV	VP.LV	VP.LV	-x-	LA.LV	VO.MT
QA.IN	-x-	—	QA.IN	VA.IN	VA.IN	VP.LV	VP.LV	-x-	LA.IN	VO.MT
LA.MT	-x-	VO.MT	LA.IN	VO.IN	LA.MT	LA.MT	VO.MT	VA.MT	-x-	VO.MT
LA.LV	LA.MT	VO.LV	LA.IN	VO.IN	LA.LV	LA.LV	VO.LV	VA.LV	-x-	VO.MT
LA.IN	-x-	VO.LV	LA.IN	VO.IN	LA.IN	LA.LV	VO.LV	VA.IN	-x-	VO.MT
VP.MT	-x-	AA.MT	AP.IN	VP.IN	VP.MT	VP.MT	VP.MT	-x-	VO.MT	VO.MT
VP.LV	VP.MT	AA.LV	AP.IN	VP.IN	VP.LV	VP.LV	VP.LV	-x-	VO.LV	VO.MT
VP.IN	-x-	AA.IN	AP.IN	VP.IN	VP.IN	VP.LV	VP.LV	-x-	VO.IN	VO.MT
AP.IN	-x-	QA.IN	QP.IN	VP.IN	VP.IN	VP.LV	VP.LV	-x-	AO.IN	VO.MT
QP.IN	-x-	—	QP.IN	VP.IN	VP.IN	VP.LV	VP.LV	-x-	QO.IN	VO.MT
VO.MT	-x-	—	AO.IN	VO.IN	VO.MT	LO.MT	LO.MT	VP.MT	-x-	VO.MT
VO.LV	VO.MT	—	AO.IN	VO.IN	VO.LV	LO.LV	LO.LV	VP.LV	-x-	VO.MT
VO.IN	-x-	—	AO.IN	VO.IN	VO.IN	LO.LV	LO.LV	VP.IN	-x-	VO.MT
AO.IN	-x-	—	QO.IN	VO.IN	VO.IN	LO.LV	LO.LV	AP.IN	-x-	VO.MT
QO.IN	-x-	—	QO.IN	VO.IN	VO.IN	LO.LV	LO.LV	QP.IN	-x-	VO.MT
LO.MT	-x-	VO.MT	AO.IN	VO.IN	VO.MT	VO.MT	VO.MT	VP.MT	-x-	VO.MT
LO.LV	LO.MT	VO.LV	AO.IN	VO.IN	VO.LV	VO.LV	VO.LV	VP.LV	-x-	VO.MT

GIP Combined Applicant and Registrar State machine

Previous research. GARP

The model by Tadashi Nakatani

- "Verification of Group Address Registration Protocol using Promela and Spin" (1997) based on IEEE 802.1p (outdated)
- The model of single segment LAN, i.e. that of one bridge (Registrar + LeaveAll) and two end stations (2 Applicants) in one LAN was considered
  - Tadashi modeled multicasting by sending messages to all the processes
  - Fast state explosion was observed



Research of Igor Konnov

Promela: sizes of processes

- GID_A: 19 control states
  - as (11 values), leave_timer (2 values), m (8 values)
  - 2 channels of mtype (8 values)
- GID_ARLA: 34 control states
  - as (11 v.), leave_timer (2 v.), m (8 v.), rs (3 v.)
  - 4 channels of mtype (8 v.)
- GIP2: 7 control states
  - 2 channels (8 v.)

Bitstate search results

- Spin has a mode to perform an approximate search using bit hash table
- Parameters: 20 hash functions, 2^{31} entries in hash table
- Results for safety checking
  - B1: depth = 10.6 M, #states = 363 M, #transitions = 862 M, time = 37 min, memory = 968 M, hash factor = 5.9
  - B2: depth = 16.2 M, #states = 19 M, #transitions = 45 M, time = 127 sec, memory = 1891 M, hash factor = 112 (very good)
  - B3: depth > 10.9 M, #states > 13 M, #transitions > 33.4 M, time = 101 sec, out of memory, hash factor = 169

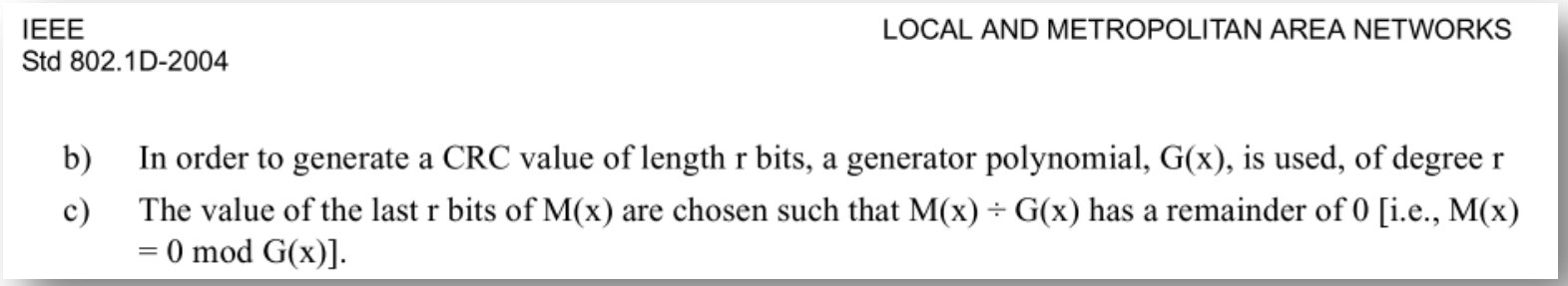
Previous research. GARP

Model of O.A Letychevskyj (2010):

380 candidates for deadlock,380 unreachability is proved by different tools. The next actions pointed to provide the complete proof was the usage of the invariant searching technique (Creation with A. Godlevsky) or the abstraction from the agent type.

Model of A.I Kotyk (2011):

5 candidates for deadlock proved to be unreachable through backward trace generation, 46 states of incompleteness. However, the model is too abstract to be sufficient to for the complete proof. For example, the requirements of the following type were not considered.



IEEE
Std 802.1D-2004

LOCAL AND METROPOLITAN AREA NETWORKS

b) In order to generate a CRC value of length r bits, a generator polynomial, G(x), is used, of degree r

c) The value of the last r bits of M(x) are chosen such that M(x) ÷ G(x) has a remainder of 0 [i.e., M(x) = 0 mod G(x)].

BPSL

- The target is to verify the labeled transition systems $< S, A, T >, T \subseteq S \times A \times S$
- **Basic protocol** can be represented as a triple $< Pre, A, Post > where\ a: pre \xrightarrow{a}$post.
- The absence of deadlocks is found through checking of completeness or the satisfiability of the disjunction
- $Pre(1) \vee Pre(2) \ldots Pre(n) = 1$
- **Predicate transformer:** Assume an assertion φ in the form of a formula of the base language means ∀s(s⊨φ)or ⊢φ in a given theory. A predicate transformer Tr(Pre,Post) is a function defined on formulae of the base language returning a new formula such that Tr(Pre,Post)→ Post. A predicate transformer strengthens the postcondition of a basic protocol by adding residual properties from the precondition: Let a specified system be a state satisfying condition γ such that γ →Pre. Therefore, the basic protocol can be applied and after it completes, condition Post will be true. As such, the basic protocol transforms Pre to Post.
- The behavior of the system: $$S_\alpha = \sum_{b \in B(\alpha)} P_b * (S_{\mathbf{Tr}(\alpha, \mathbf{post}(b))} + \Delta)$$

BPSL

The base language of the BPSL relies on attribute representation of Moore Automata $< S, A, U, T, \phi >, \phi: S \to U, where\ U = D^R$ - set of agent attributes.

The agent is determined by it's set of behaviors, the external environment of the agent is the agent E with the insertion function: $< E, C, A, Ins > \{1\}$, where *E* is the set of the states of the environment, *C* is the set of actions of the environment, *A* is the set of actions of the agents towards the environment and *Ins* is the insertion function.

$$Ins: E \times F(A) \to E\ \{2\}\ \ Ins(e,u) = e[u]\ \ e[u]_E\ \{3\}$$

$$e[u,v] = (e[u])[v]\ \{4\}$$

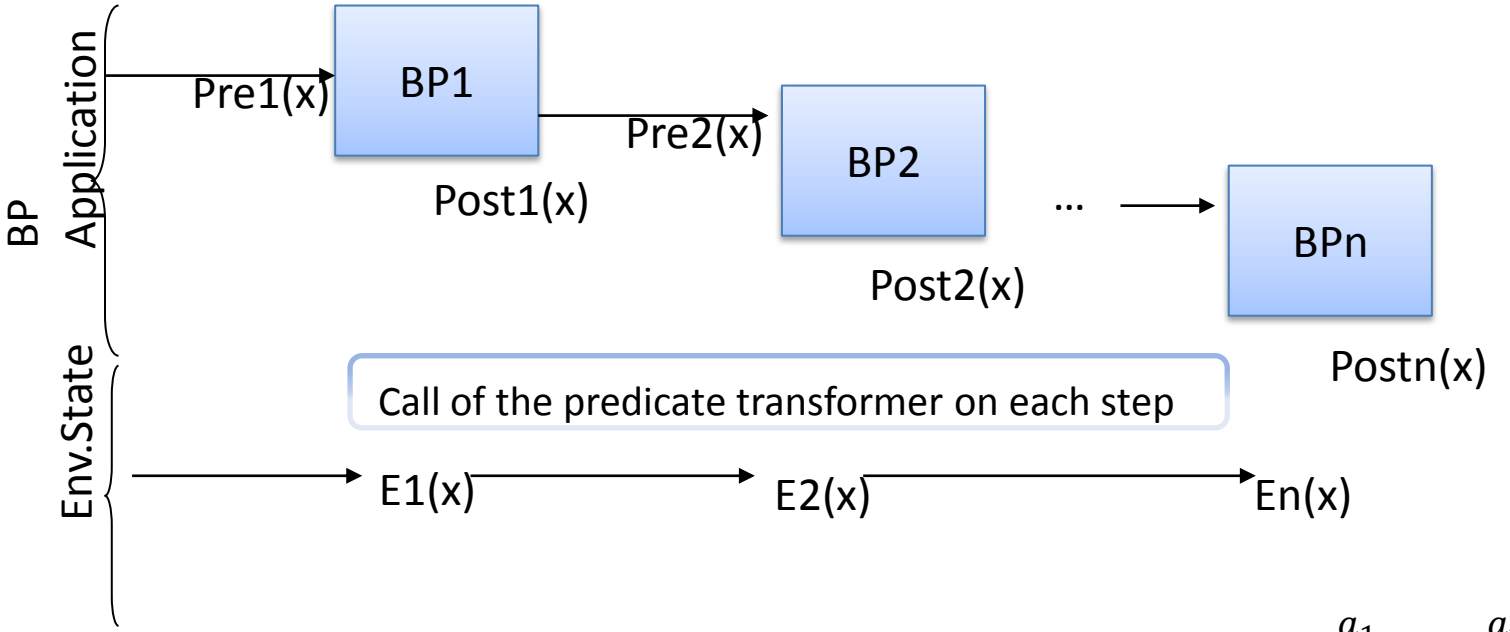
$$e[u_1,u_2,...], e'[e[u_1,u_2,...]_E,...]_{E'}$$

The transmissions of the messages is represented as $\mathrm{Out}\left(T\ a, T_1 a_1,\ x(p_1 \dots p_m)\right)$.

The incoming message is given as $In(T\ a, z)$

BPSL

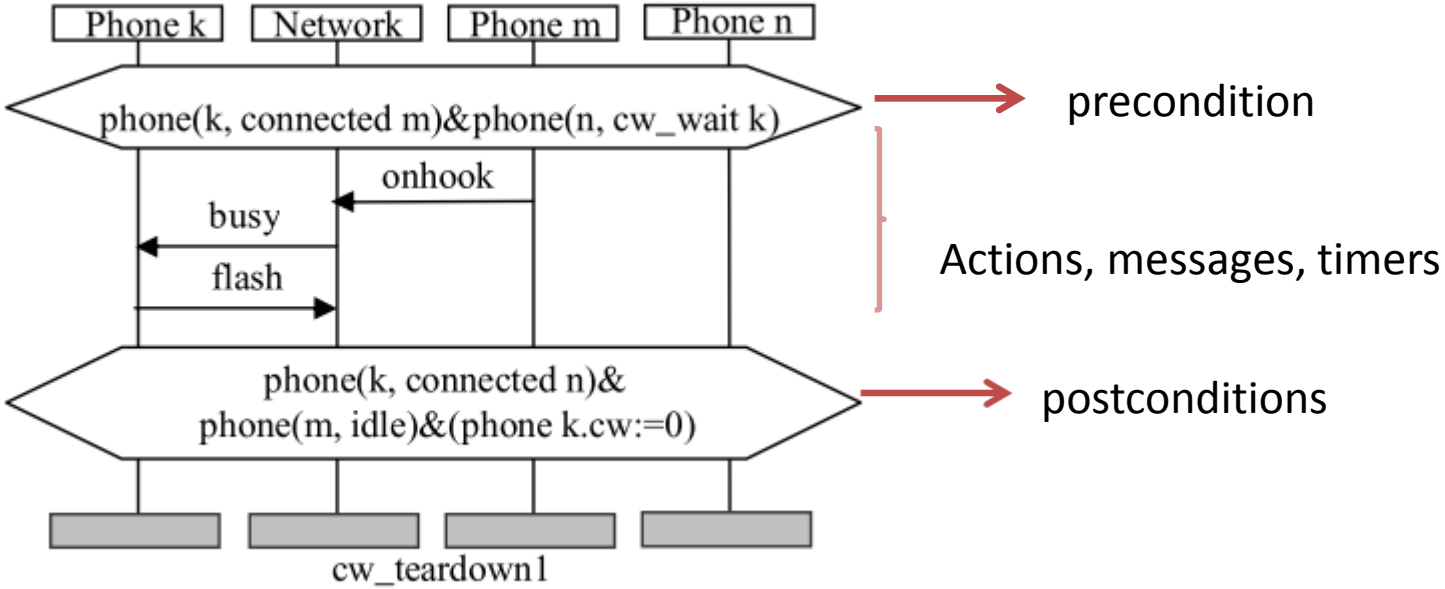
There are several tools in the VRS client, such as Static Requirement Checker(is an actual prover), Concrete Trace Generator(is a model checker), Symbolic Trace Generator. The symbolic trace generator applies basic protocols as follows:



Trace is formed by sequence of changing of state of environment $E_0 \xrightarrow{a_1} E_1 \ldots \xrightarrow{a_n} E_n \ldots \xrightarrow{a_j}$

Enlarging technique

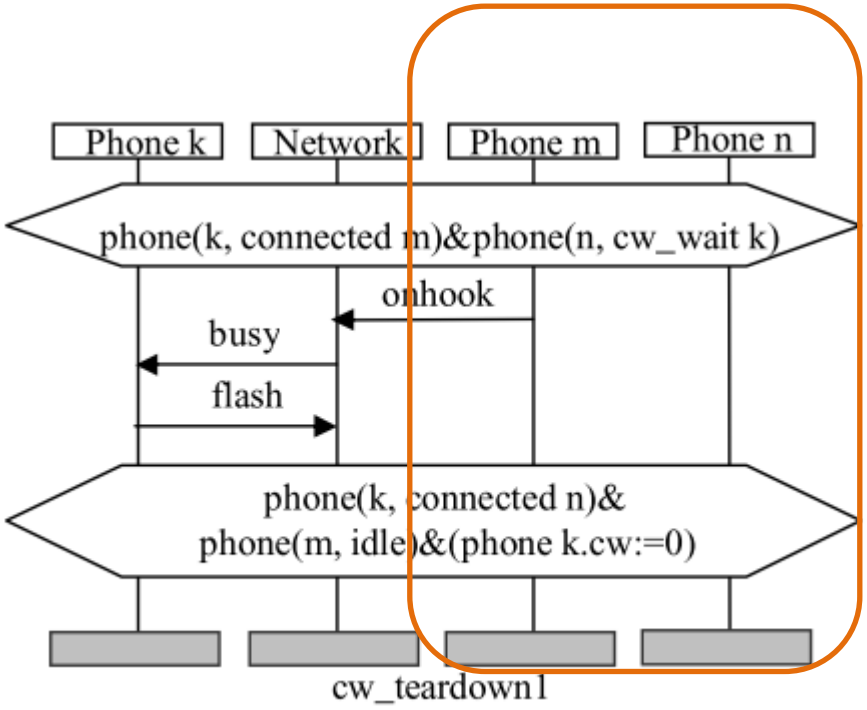
Representation of BPSL through MSC diagrams



We use the state assertion state(x,m,s) as an atomic formula describing that the agent with name m has the type x and is in a state s at the current moment of time (as shorthand we use the notation x(m,s)). An attribute r of agent m of a type x is denoted as xm.r; if this attribute has parameters t1,...,tn, it will be denoted as xm.r(t1,...,tn)

Enlarging technique

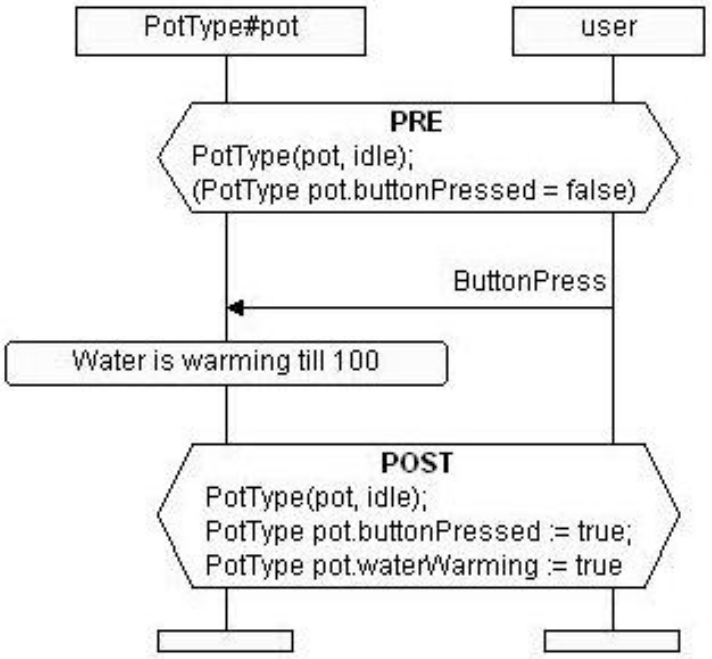
- Representation of BPSL through MSC diagrams

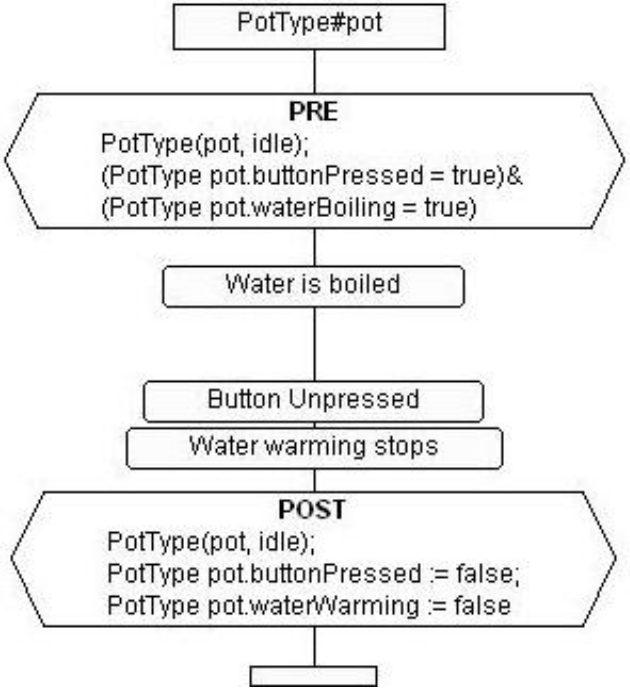


Under certain conditions different agents can be aggregated into one Group of the agent creating the separate functionality interacting with remaining agents. This is the core of Enlarging technique.
The tool for this method is on the developing stage.

Example of the project

- Boiler has one button;
- After pushing this button it shall heat the water to 100 degrees;
- After reaching boiling the button should be pushed out automatically and the heating - stopped.





Example of the project

Let us consider 1 trace in STG. There is no initialization, thus buttonPressed $\in (true, false)$; waterBoiling $\in (true, false)$

1) **APPLICABLE BP1**

$$\frac{\text{Pre: buttonPressed= false}}{\text{Post:buttonPressed=true}};$$

$\longrightarrow$ 2) **APPLICABLE BP2**

$$\frac{\text{Pre: buttonPressed= true\&waterBoiling=true}}{\text{Post:buttonPressed=false\&waterBoiling=false}}$$

3) **APPLICABLE BP1**

$$\frac{\text{Pre: buttonPressed= false}}{\text{Post:buttonPressed=true}};$$

NO APPLICABLE BP REMAINED

$\longrightarrow \perp$

In STG you define trace which can be created by this set of basic protocols

Industrial Application

The Enlarging technique with Symbolic Modeling of Complex Functions would give an opportunity to verify low – level requirements of the agent systems for any type of engineering specifications.

For example, the specifications for NASA Small Aircraft Transportation System contained the safety functions, which were verified as a conjunction of such properties:

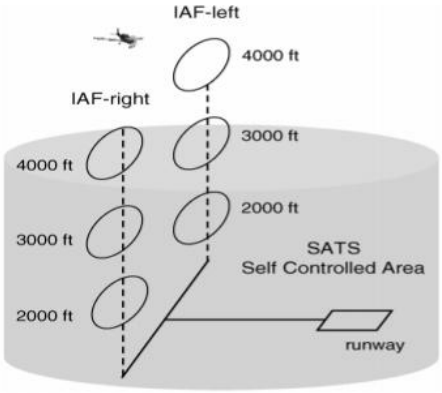
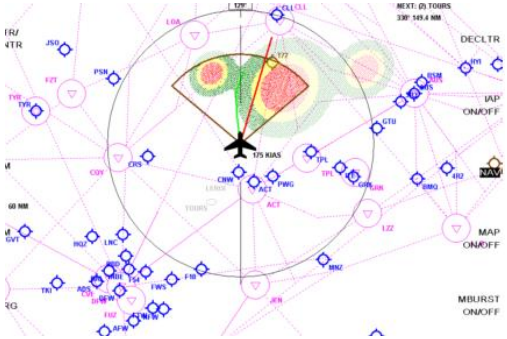


Figure 1. SATS SCA model.



1. There are no more than two aircraft assigned to a MAHF (left or right).
2. The number of aircraft inside the SCA at an IAF (left or right) is less than or equal to two.
3. There is at most one aircraft at hold 2000 (left or right) and at hold 3000 (left or right).
4. There are no more than two aircraft at the miss approach zone (left or right).
5. When an aircraft is in the lateral entry (left or right) there are no aircraft in hold 3000, hold 2000 or miss approach zone (left or right), respectively.

Industrial Application

However, the requirement such as “When an aircraft is in the lateral entry (left or right) there are no aircraft in hold 3000, hold 2000 or miss approach zone (left or right), respectively” is not concrete, as there was determined the function constraining the interaction between agents which has to be calculated as follows.

When the distance between agent i and j is less than a defined value, repulsive field is associated with each agent to prevent collision. The repulsive field is:

$$U_r(x_i, x_j) = \begin{cases} -\frac{1}{2\delta_{ij}}(r_{ij} - (r_j + \delta_{rj}))^2 & if\ r_j \le r_{ij} \le r_j + \delta_{rj} \\ 0 \end{cases}$$

$$F_r(x_i, x_j) = \nabla U_r(x_i, x_j) = \frac{1}{\delta_{ij}}\left(\frac{r_j + \delta_{rj}}{r_{ij}} - 1\right)\begin{bmatrix} x_i - x_j \\ y_i - y_j \end{bmatrix}$$

The force on agent i is

$$F_i = F_r(x_i, x_j) + F_a$$

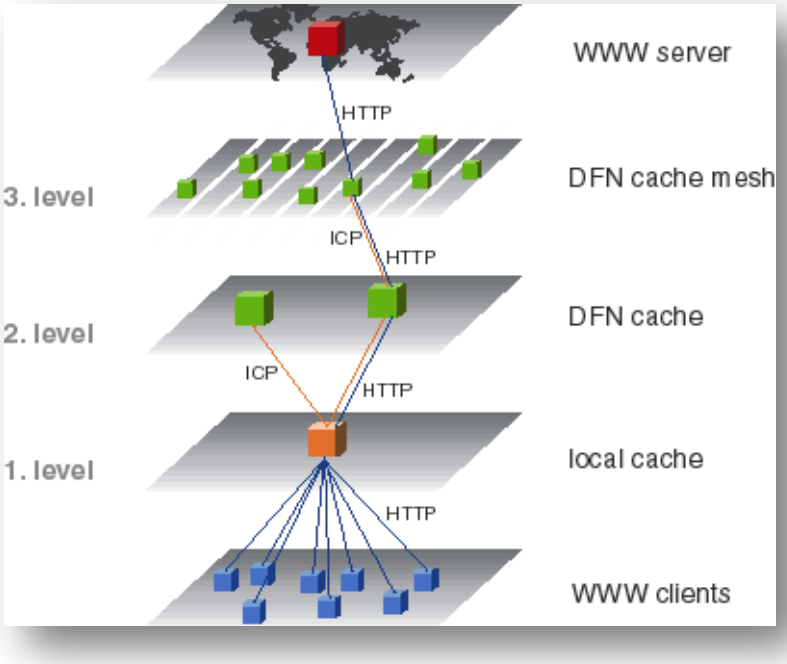
In previous research such requirements were neglected however, they were usually estimated during the simulation, failures in the Safety Critical Systems such as Aircraft Systems can cause human lives.

Formulation of the problem

Let us assume that the system contains several enlarged entities.

In the example of the Hierarchical caches L2 represents the “Enlarged” entity

DFN cache which contains 2 different agent types – ICP and HTTP which correspond to Local clients.



Thus, the task is to

1) build a formal model of a Transition system with enlarged entities

2) Check whether the safety property which is proved in the model without Enlarging corresponds to the one Proved with Enlarging. The Enlarging technique is not correct for the general case and classes or tasks for which this technique is advantageous are not yet determined, thus we have to work on the set of examples with adequate scenarios.

3) Refine the checker with the case of non-linearity.

Formulation of the problem

In example of creation of the scenario:

$$Pre(Initial) \rightarrow Pre(First\ Satisfiable) \rightarrow Effect(Post) \rightarrow Assignment(Predicate\ Transformer) \rightarrow Pre(First\ Satisfiable) \rightarrow Termination/Failure.$$

If Pre and Post syntax is enriched with Polynomial functions, Assignment remains the same, the following should be added:

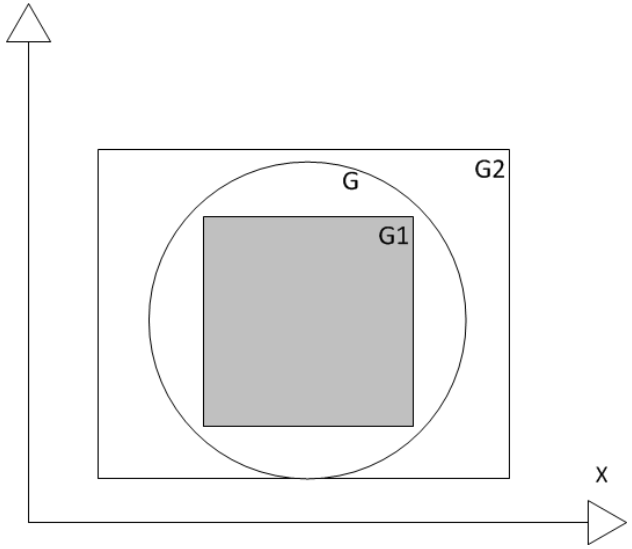
1) Checking the satisfiability of Pre
2) Methods of approximation for the PT

For example:

If $(x^2+y^2 = 1) \in G$

We can construct G':

$(a \leq x \leq b \wedge c \leq y \leq d) \in G'$



Future Program

1. What G′to choose $G \Rightarrow G' or G' \Rightarrow G$ to satisfy the following properties?

Safety: $(\forall \sigma: \sigma \epsilon S^{\omega}: \sigma \vDash P \Leftrightarrow (\forall\, i: i \geq 0: (\exists \beta \in S^{\omega}: \sigma[\ldots i]\beta \vDash P)))$;

Where S^{ω}- set of infinite sequence of states, P- program states, $\sigma \nvDash P$ – a bad prefix, σ is an infinite sequence of prefixes.

Liveness: $\forall \alpha: \alpha \in S^{*}: (\exists \beta: \beta \in S^{\omega}: \alpha\beta \vDash P)$, where S^{*} is the finite set of finite sequences of states.

2. Are there any algorithms for checking satisfiability which would have the complexity less exponential?

3. Should we use standard approximation algorithms such as Artificial Bee Colony and how to prove their correctness for this case?

References

- Alexander A. Letichevsky: Basic Protocols: Specification Language for Distributed Systems. Ershov Memorial Conference 2006: 21-25
- Alexander A. Letichevsky, Julia V. Kapitonova, A. A. Letichevsky Jr., Vladislav A. Volkov, Sergey Baranov, Thomas Weigert: Basic protocols, message sequence charts, and the verification of requirements specifications. Computer Networks 49(5): 661-675 (2005)
- Alexander A. Letichevsky, David Gilbert: A Model for Interaction of Agents and Environments. WADT 1999: 311-328
- Igor Konnov, O.A. Letichevsky Jr. Model Checking GARP Protocol using Spin and VRS. International Workshop on Automata, Algorithms, and Information Technologies. Kiev, May, 2010.
- Agarwal, P. and Manuel, L. (2008). Extreme loads for an offshore wind turbine using statistical extrapolation from limited field data. Wind Energy11 673-684.
- Bai, Chongyang; Zhang, Xuejun; , "Aircraft Landing Scheduling in the Small Aircraft Transportation System," *Computational and Information Sciences (ICCIS), 2011 International Conference on* , vol., no., pp.1019-1022, 21-23 Oct. 2011
doi: 10.1109/ICCIS.2011.65
- Apern B., Schneider F.B. Defining liveness. Information procession letters 21(Oct. 1985),p 181-185.