Updatable Strategy Logic

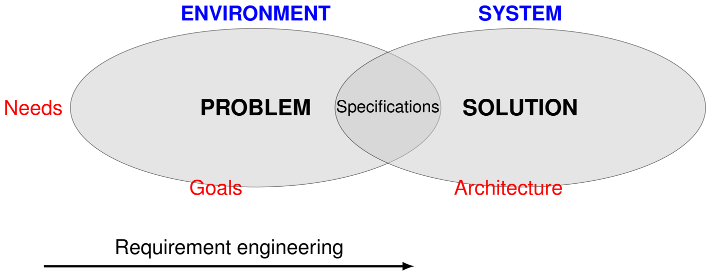
Christophe Chareton   Julien Brunel   David Chemouil

Onera, Toulouse

January 8, 2013

Outline

Requirements engineering



- Identify requirements
- Derive specifications from functional requirements

KHI

- KHI in three sentences:
  - The progressive goals refinement leads to specifications that are expressed in LTL.
  - These specifications are gathered into *roles* (LTL).
  - We focus on the problem of a possible assignment of those roles to coalitions of agents.
- Main stakes:
  - Provide sets of specifications that are structured by the agents that have to ensure them.
  - Identify those of these specifications that we cannot ensure with the provided agents.

Formalism, a first approach with Alternating-Time Temporal logic (ATL: Alur, Henzinger, Kupferman)

- Problem:
  - A set $\mathcal{R}$ of roles and a set Σ of actors.
  - An assignment relation $\subseteq \mathcal{R} \times \Sigma$.
  - Question: for all role $r \in \mathcal{R}$, are the concerned agents able to ensure r (LTL)?
- ATL
  - ATL enables to express properties of capabilities of agents to ensure temporal properties.
  - $$\langle\langle A \rangle\rangle \varphi$$
  - Agents in coalition A are able to ensure the satisfaction of property expressed by φ **whatever the other agents do**.

Problems met

- Take into account the interaction between coalitions
  - Two roles r_1 and r_2, two coalitions A_1 and A_2.
  - A_1 can ensure r_1 but A_2 cannot ensure r_2.
  - Is A_1 able to ensure its role and to enable A_2 to ensure its role at the same time?
  - Not expressible in ATL

$$\langle\langle A_1 \rangle\rangle (r_1 \wedge \langle\langle A_2 \rangle\rangle r_2)$$

- An agent may be part of several coalitions:
  - If $A_1 \cap A_2 \neq \emptyset$, then how to express that A_1 and A_2 can ensure their respective roles by playing along a non-contradictory strategy?

$$\langle\langle A_1 \rangle\rangle r_1 \wedge \langle\langle A_2 \rangle\rangle r_2$$

Strategy Logic (SL: Mogavero, Murano, Perelli, Vardi)

- An observation:

$$\langle\langle A \rangle\rangle \varphi$$

  There is a strategy x such that if A plays along x then φ is ensured.
- Starting idea for SL: separate both elements:
  - A quantifier $\langle\langle x \rangle\rangle$: $\langle\langle x \rangle\rangle \varphi$ is true iff there is a strategy x such that φ is ensured.
  - A strategy binder (A, x): $(A, x)\varphi$ is true iff if A plays along strategy for x then φ is ensured.
- Sub-formulas are evaluated in **contexts that stores the quantifiers and binders**.
- At evaluation of temporals, each agent is bound to a strategy.
- Enables to treat the first problem:

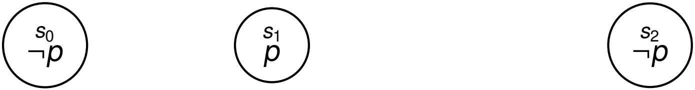
$$\langle\langle x_1 \rangle\rangle (A_1, x_1)([\![x_2]\!](\Sigma \backslash A_1, x_2)(r_1 \wedge \langle\langle x_3 \rangle\rangle (A_2, x_3)(r_2))$$

- The second one still holds ...

Semantics of SL: *CGS*

Concurrent Game Structures:

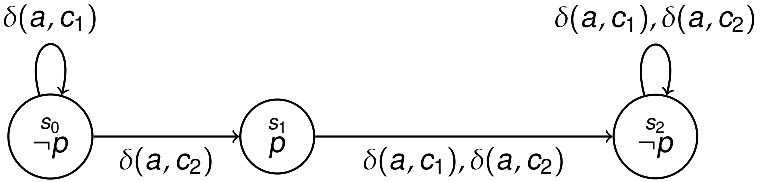
- Some elements from classical Kripke structures:
  - A set of states M
  - A set of atomic propositions At
  - A valuation function, from M to $\mathcal{P}(\mathrm{At})$
- Transitions:
  - A set of agents Σ
  - A finite set of possible actions for the agents $A \subsetneq \mathbb{N}$
  - In each state, each agent plays a choice and the transitions are determined by the expressed actions : δ is a function from $M \times A^{\Sigma}$ to M.



Semantics of SL: *CGS*

Concurrent Game Structures:

- Some elements from classical Kripke structures:
  - A set of states M
  - A set of atomic propositions At
  - A valuation function, from M to $\mathcal{P}(\mathrm{At})$
- Transitions:
  - A set of agents Σ
  - A finite set of possible actions for the agents $A \subsetneq \mathbb{N}$
  - In each state, each agent plays a choice and the transitions are determined by the expressed actions : δ is a function from $M \times A^{\Sigma}$ to M.



Semantics of SL: quantifiers and binders

- A *strategy* is a function σ from M^* to A
- A *context* κ maps agents and strategy variables to strategies.

Definition

Satisfaction

- $\mathcal{M}, \kappa, s \models_{SL} \langle\langle x \rangle\rangle \varphi$ iff there is a strategy σ such $\mathcal{M}, \kappa[x \to \sigma], s \models_{SL} \varphi$
- $\mathcal{M}, \kappa, s \models_{SL} (a, x)\varphi$ iff $\mathcal{M}, \kappa[a \to \kappa(x)], s \models_{SL} \varphi$

where $\kappa[a \to \sigma]$ is obtained from κ by **replacing its value for *a* with σ.**

SL uses contexts that do not enable to compose several strategies for an agent

USL: main ideas

- In SL: when a binder (A, x) occurs, current strategy for A is automatically revoked.
- Aims:
  - either update current strategy without revoking it.
  - either revoke it.
- Means:
  - In general case, a binder $(A \triangleright x)$ does not delete the strategies previously bound to A.
  - We make explicit the, perhaps, revocation of strategy: introduction of an unbinder $(A \not\triangleright x)$ expressing it.
  - Delete the constraint for temporals only under complete context.
- Observation: The SL binder (A, x) again is decomposed into two operations
  - **Agents in A are unbound from their current strategies.**
  - They are bound to strategy instanciating x.

Syntax

Definition

Let Σ be a set of agents, At a set of propositions and X a set of variables, $\mathrm{USL}(\Sigma, \mathrm{At}, X)$ is given by the following grammar:

- State formulas:

$$\varphi ::= p \mid \neg\varphi \mid \varphi \wedge \varphi \mid \langle\!\langle x \rangle\!\rangle \varphi \mid (A \triangleright x)\psi \mid (A \not\triangleright x)\psi$$

- Path formulas:

$$\psi ::= \varphi \mid \neg\psi \mid \psi \wedge \psi \mid \psi \,\mathbf{U}\, \psi \mid \circ\psi$$

where $p \in \mathrm{At}, A \subseteq \Sigma, x \in X$.

Closed formulas are evaluated with no context.

Semantics: progression of the presentation

Semantics:

- Structures (*NATS*)
- Adaptation of the notion of contexts: strategies and plans
- Plan transformations
- Satisfaction relation

Semantics: *NATS*

Definition

A Non-deterministic Alternating Transition System (*NATS*) is a tuple $\mathcal{M} = \langle \Sigma, M, \mathrm{At}, \pi, \delta \rangle$ where:

- A set M of states, a set At of atomic propositions, a valuation function π, from M to $\mathcal{P}(\mathrm{At})$, a set Σ of agents.
- A transition function $\delta : \Sigma \times M \rightarrow \mathcal{P}(\mathcal{P}(M))$. It maps a pair $\langle agent, state \rangle$ to a non-empty family of choices of possible next states.

- Choices depend on states and agents.
- δ directly gives the sets of potential successor.

Semantics: Strategies and plans

Definition

- A *strategy* is a function σ from $\Sigma \times M^*$ to $\mathcal{P}(M)$ such that for all $(a, \tau) \in \Sigma \times M^*, \sigma(a, \tau) \in \delta(a, last(\tau))$.
- A *memory* μ is a partial function from X to $Strat$, storing the instantiations for quantified strategies.
- A *context* κ is a finite list of pairs in $(\mathcal{P}(\Sigma) \times X)$, representing the structure of the active bindings.
- A *plan* Π is a pair of a memory and a context. A plan induces a function from M^* to $\mathcal{P}(M)$: $(\mu, (A, x))(\tau) = \mu(x)(A, \tau)$ and $(\mu, \kappa \cdot (A, x))(\tau) =$
  - $(\mu, \kappa)(\tau) \cap \mu(x)(A, \tau)$ iff it is not empty,
  - else $(\mu, \kappa)(\tau)$

Semantics: Plan transformations

The semantics also uses the following transformations for a context:

- A plays x : $\kappa[A \rightarrow x] = \kappa \cdot (A, x)$
- A revokes x:
  - $(A_1, x)[A \nrightarrow x] = (A_1 \backslash A, x)$
  - $(\kappa \cdot (A_1, x))[A \nrightarrow x] = \kappa[A \nrightarrow x](A_1 \backslash A, x)$
- Quantifier:
  - for all x_i in $dom(\mu) \backslash \{x\}, \mu[x \rightarrow \sigma](x_i) = \mu(x_i)$
  - $\mu[x \rightarrow \sigma](x) = \sigma$.

Semantics: satisfaction

Definition

Let $\mathcal{M}$ be a *NATS*, then for all memory μ, context κ and state s,

- $\mathcal{M}, \mu, \kappa, s \models \langle\langle x \rangle\rangle \varphi$ iff there is a strategy $\sigma \in \mathit{Strat}$ such that $\mathcal{M}, \mu[x \to \sigma], \kappa, s \models \varphi$
- $\mathcal{M}, \mu, \kappa, s \models (A \rhd x)\varphi$ iff for all λ in $\mathit{out}(\mu, \kappa[A \to x]), \mathcal{M}, \mu, \kappa[A \to x], \lambda \models \varphi$
- $\mathcal{M}, \mu, \kappa, s \models (A \not\rhd x)\varphi$ iff for all λ in $\mathit{out}(\mu, \kappa[A \nrightarrow x]), \mathcal{M}, \mu, \kappa[A \nrightarrow x], \lambda \models \varphi$

Let φ be a closed formula, then $\mathcal{M}, s \models \varphi$ iff $\mathcal{M}, \mu_\emptyset, \kappa_\emptyset \models \varphi$.

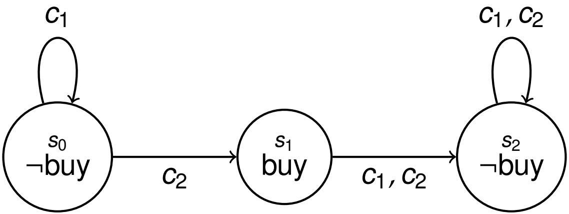
The second problem from Khi is resolved:

If $A_1 \cap A_2 \neq \emptyset$, can A_1 and A_2 ensure their respective roles by playing along a non-contradictory strategy?

$$\langle\langle x_1 \rangle\rangle (A_1 \rhd x_1)(r_1 \wedge \langle\langle x_2 \rangle\rangle (A_2 \rhd x_2) r_2)$$

Expressive power: Sustainable capability

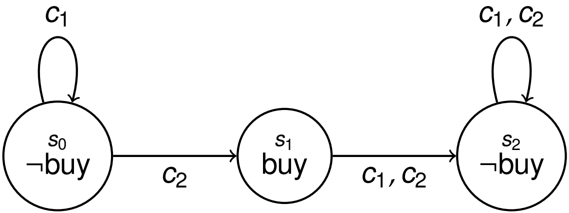
- A notion of *sustainable capabilities*:
- A capability for an agent that remains active even if already employed.
- Intuitive example: *A*lice can always buy car
  - She can buy a car once and decide when
  - In SL: $\langle\langle x_1\rangle\rangle(a, x_1)\Box(\langle\langle x_2\rangle\rangle(a, x_2) \circ \text{buy})$
  - In USL: $\langle\langle x_1\rangle\rangle(a \rhd x_1)\Box(\langle\langle x_2\rangle\rangle(a \not\rhd x_1)(a \rhd x_2) \circ \text{buy})$



- True at s_0 by strategy *allways*-c_1
- She can remain able to buy it, but only provided she never does.
- Her capability to buy a car is not sustainable.

Expressive power: sustainable capability

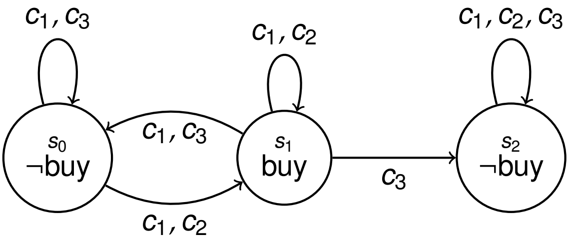
- Intuitive example: *Alice* can always buy car
  - She can buy as many as she wants whenever she wants:
  - In USL: $\langle\langle x_1 \rangle\rangle (a \triangleright x_1) \Box (\langle\langle x_2 \rangle\rangle (a \triangleright x_2) \circ \text{buy})$



- false at s_0 since contraditory strategies.

Expressive power: sustainable capability

- Intuitive example: *Alice* can always buy car
  - She can buy as many as she wants whenever she wants:
  - In USL: $\langle\langle x_1 \rangle\rangle (a \triangleright x_1) \Box (\langle\langle x_2 \rangle\rangle (a \triangleright x_2) \circ \mathrm{buy})$



- true at s_0:
  - any occurence of c_2 from s_0 or s_1 buys a car.
  - *always*-c_1 enables to maintain the capability.
  - *always*-c_1 is not contradictory with any occurence of c_2

Expressive power: results

Theorem

There is a transformation of CGS $\mathcal{G}'$ to NATS $\mathcal{G}'$ and from formulas θ in SL to formulas θ' in USL such that for all $\theta \in$ SL and for all CGS $\mathcal{G}$, $\mathcal{G} \models \theta$ iff $\mathcal{G}' \models \theta'$. Furthermore, upon SL$\{1\}$, this transformation reduces to the actions-choices equivalence.

Theorem

There is a formula in USL$\{1\}$ not expressable in SL$\{1\}$.

We proved the second theorem with formula $\Gamma_\infty := \langle\langle x \rangle\rangle (a \rhd x) \Box (\langle\langle y \rangle\rangle (a \rhd y) \circ p \wedge \langle\langle y \rangle\rangle (a \rhd y) \circ \neg p)$. It asserts that a is sustainably able to decide whether p holds or not in next state.

Model-checking: results

Theorem

- *The model-checking of USL is* ***NONELEMENTARYTIME*** *decidable.*
- *The model-checking of USL under memoryless strategies (USL^0) is* ***PSPACE****-complete.*

Conclusion

A formalism that:

- Enables composition of strategies for one agent and the sustainable capabilities.
- Unifies it with the classical branching-time mechanisms of strategies' revocation.
- Uses strategies that are both updatable and revocable.
- Holds similar model-checking results as comparable formalisms (SL, ATL_{sc}, Brihaye, Da Costa, Laroussinie, Markey)

Future works

- Expressive power:
  - Express sustainable capabilities as fixed points properties, compare USL with extensions of μ–calculus dealing with strategies (QD_μ, S. Pinchinat).
  - Further explore the possibilities enabled by free use of the unbinder.
- Related to KHI:
  - Further criteria for model correctness: ensure a role *rl* assigned to an actor *a* does not contradict its pursued goals.
  - Compare the efficiency of different strategies in case they do not fully ensure the satisfaction of the roles.

Thank you for your attention

Any question?