Coco: Runtime Reasoning About Conflicting Commitments

Nirav Ajmeri, Jiaming Jiang, Rada Chirkova, Jon Doyle, Munindar P. Singh

Department of Computer Science
North Carolina State University

July 15, 2016

Outline

Introduction

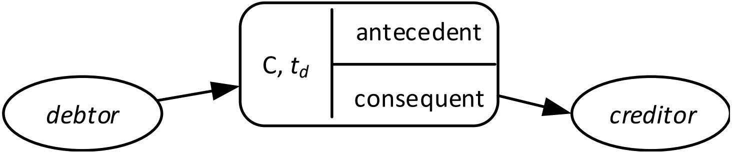
Formal Language

Formalizing Healthcare Scenario

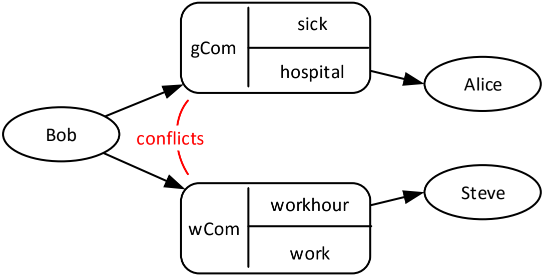
Conclusion

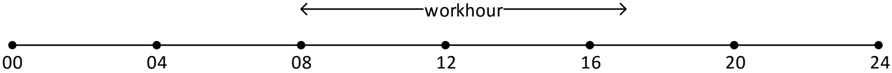
Commitment Schema

- For effective interactions, agents enter into commitments



Commitment Instances





Problem

How to detect and resolve runtime conflicts?

Contributions

- Representations for commitments and dominance
- Decision procedures for determining compliance of actions with commitment instances
    - apply ASP to identify nondominated commitments
    - apply Alechina et al.'s [2013] techniques to determine compliance of agent actions with nondominated commitment instances

Outline

Introduction

Formal Language

Formalizing Healthcare Scenario

Conclusion

An Example Commitment Schema

Definition of the gCom schema

1 $schema(gCom)$

2 $guardianR(C, G, T) \rightarrow dbt(gCom, G, T)$

3 $guardianR(C, G, T) \rightarrow crd(gCom, C, T)$

4 $[sick(C, T) \wedge crd(gCom, C, T)] \rightarrow ant(gCom, T)$

5 $[bring(G, C, Ped, T) \wedge pedR(C, Ped, T){\wedge}dbt(gCom, G, T) \wedge crd(gCom, C, T)]$
$\rightarrow con(gCom, T)$

6 $dDuration(gCom, 3)$

Schema Predicates

Schema predicates	
$schema(S)$	$ant(S, T)$
$dbt(S, P, T)$	$con(S, T)$
$crd(S, P, T)$	$dDuration(S, T)$
Instance predicates	
$sInst(Si)$	$violated(Si, T)$
$isInstOf(Si, S)$	$satisfied(Si, T)$
$dbtI(Si, P)$	$becomesDetached(Si, T)$
$crdI(Si, P)$	$becomesViolated(Si, T)$
$antI(Si, T)$	$becomesSat(Si, T)$
$conI(Si, T)$	$conflicting(Si_1, Si_2, T)$
$dDurationI(Si, T)$	$dominates(Si_1, Si_2, T)$
$detached(Si, T)$	$dominated(Si, T)$
$sameDbtI$	$conINotHold(Si, T_1, T)$
$sameCrdI$	

Satisfaction and Violation

Satisfaction

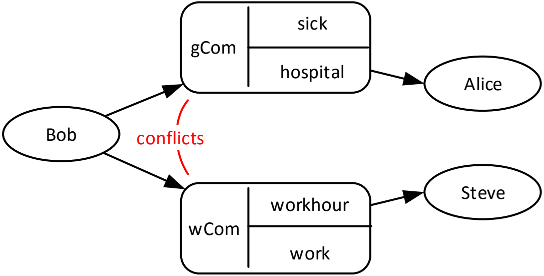
1 $[\mathit{detached}(\mathit{Si}, T') \land \mathit{conI}(\mathit{Si}, T) \land \mathit{difference}(T, T', \mathit{1})] \rightarrow \mathit{becomesSat}(\mathit{Si}, T)$

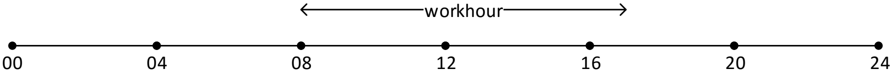
Violation

1 $[\mathit{becomesDetached}(\mathit{Si}, T_{\mathit{1}}) \land \mathit{dDurationI}(\mathit{Si}, T_{\mathit{2}})$
$\land \mathit{addition}(T_{\mathit{1}}, T_{\mathit{2}}, T) \land \mathit{conINotHold}(\mathit{Si}, T_{\mathit{1}}, T)$
$\land \neg \mathit{dominated}(\mathit{Si}, T)] \rightarrow \mathit{becomesViolated}(\mathit{Si}, T)$

Conflict between Commitment Instances

Based on Semantics

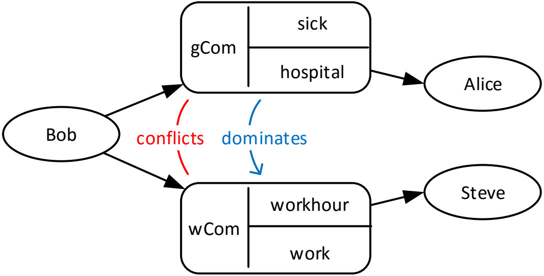


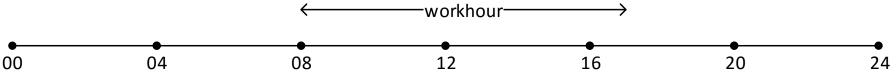


1 $[detached(Si_1, T) \wedge detached(Si_2, T) \wedge \neg(conI(Si_1, T) \leftrightarrow conI(Si_2, T))] \rightarrow conflicting(Si_1, Si_2, T)$

Dominance between Commitment Instances

Based on Preferences





$$[\mathit{isInstOf}(Si_1, gCom) \wedge \mathit{isInstOf}(Si_2, wCom) \wedge \mathit{detached}(Si_1, T) \wedge \mathit{detached}(Si_2, T) \wedge \mathit{sameDbtI}(Si_1, Si_2)] \rightarrow \mathit{dominates}(Si_1, Si_2, T)$$

Outline

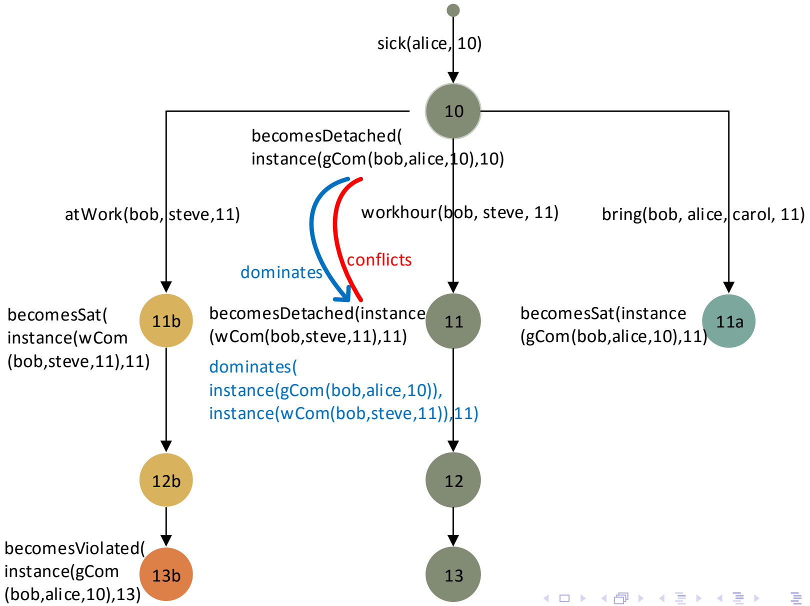
Introduction

Formal Language

Formalizing Healthcare Scenario

Conclusion

Possible Enactments


sick(alice, 10)
10
becomesDetached(instance(gCom(bob,alice,10),10)
atWork(bob, steve,11)
workhour(bob, steve, 11)
bring(bob, alice, carol, 11)
dominates
conflicts
becomesSat(instance(wCom (bob,steve,11),11)
11b
becomesDetached(instance (wCom(bob,steve,11),11)
11
becomesSat(instance (gCom(bob,alice,10),11)
11a
dominates(instance(gCom(bob,alice,10)), instance(wCom(bob,steve,11)),11)
12b
12
becomesViolated(instance(gCom (bob,alice,10),13)
13b
13

Conflict-detection, Formally

1 $[\mathit{atLocation}(P, L_1, T) \wedge (L_1 \neq L_2)] \rightarrow \neg \mathit{atLocation}(P, L_2, T)$

2 $[\mathit{isInstOf}(\mathit{Si}, \mathit{gCom}) \wedge \mathit{conI}(\mathit{Si}, T) \wedge \mathit{dbtI}(\mathit{Si}, G)] \rightarrow \mathit{atLocation}(G, \mathit{hospitalLoc}, T)$

3 $[\mathit{employerR}(\mathit{Ee}, \mathit{Er}, T) \wedge \neg \mathit{atLocation}(\mathit{Ee}, \mathit{companyLoc}, T)] \rightarrow \neg \mathit{atWork}(\mathit{Ee}, \mathit{Er}, T)$

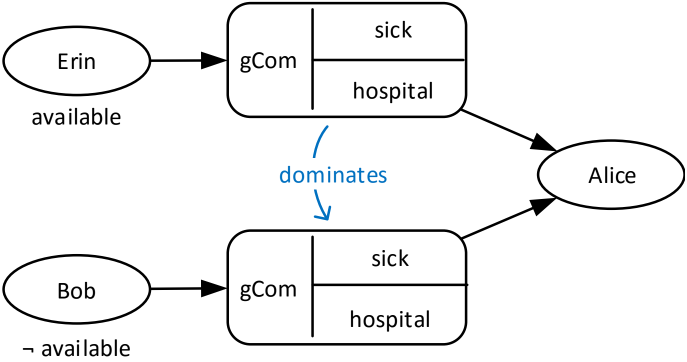
4 $[\mathit{isInstOf}(\mathit{Si}, \mathit{wCom}) \wedge \mathit{dbtI}(\mathit{Si}, \mathit{Ee}) \wedge \mathit{crdI}(\mathit{Si}, \mathit{Er})$
$\wedge \neg \mathit{atWork}(\mathit{Ee}, \mathit{Er}, T)] \rightarrow \neg \mathit{conI}(\mathit{Si}, T)$

5 $[\mathit{isInstOf}(\mathit{Si}, \mathit{wCom}) \wedge \mathit{conI}(\mathit{Si}, T) \wedge \mathit{dbtI}(\mathit{Si}, \mathit{Ee})$
$\wedge \mathit{crdI}(\mathit{Si}, \mathit{Er})] \rightarrow \mathit{atLocation}(\mathit{Ee}, \mathit{company}, T)$

6 $[\mathit{guardianR}(C, G, T) \wedge \mathit{pedR}(C, \mathit{Ped}, T) \wedge \neg \mathit{atLocation}(G, \mathit{hospitalLoc}, T)]$
$\rightarrow \neg \mathit{bring}(G, C, \mathit{Ped}, T)$

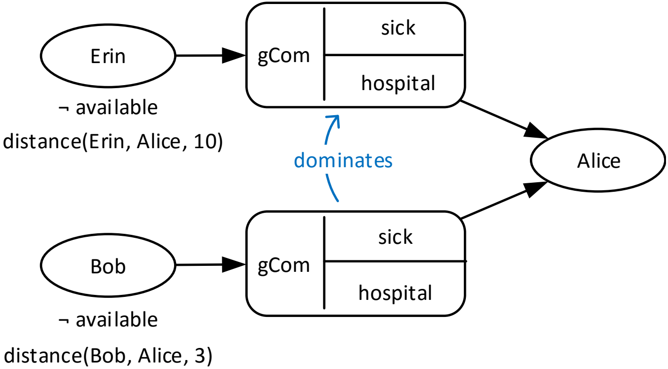
7 $[\mathit{isInstOf}(\mathit{Si}, \mathit{gCom}) \wedge \mathit{dbtI}(\mathit{Si}, G) \wedge \mathit{crdI}(\mathit{Si}, C)$
$\wedge \mathit{pedR}(C, \mathit{Ped}, T) \wedge \neg \mathit{bring}(G, C, \mathit{Ped}, T)] \rightarrow \neg \mathit{conI}(\mathit{Si}, T)$

Example Basis for Dominance: Availability



1 $[\mathit{isInstOf}(Si_1, gCom) \wedge \mathit{isInstOf}(Si_2, gCom) \wedge \mathit{detached}(Si_1, T) \wedge \mathit{detached}(Si_2, T)$
$\wedge \mathit{sameCrdl}(Si_1, Si_2) \wedge \mathit{dbtl}(Si_1, G_1) \wedge \mathit{dbtl}(Si_2, G_2) \wedge (G_1 \neq G_2)$
$\wedge \mathit{available}(G_1, T) \wedge \neg \mathit{available}(G_2, T)] \rightarrow \mathit{dominates}(Si_1, Si_2, T)$

Example Basis for Dominance: Proximity



1 $[isInstOf(Si_1, gCom) \wedge isInstOf(Si_2, gCom) \wedge detached(Si_1, T) \wedge detached(Si_2, T)$
$\wedge sameCrdl(Si_1, Si_2) \wedge crdl(Si_1, C) \wedge dbtl(Si_1, G_1) \wedge dbtl(Si_2, G_2) \wedge (G_1 \neq G_2)$
$\wedge \neg available(G_1, T) \wedge \neg available(G_2, T) \wedge dist(G_1, C, D_1, T) \wedge dist(G_2, C, D_2, T)$
$\wedge (D_1 \leq D_2)] \rightarrow dominates(Si_1, Si_2, T)$

Outline

Introduction

Formal Language

Formalizing Healthcare Scenario

Conclusion

Conclusion and Future Directions

Coco provides

- Flexible formalization of instances
- Reasoning about conflicts between commitments and dominance at runtime

Subsequent work

- Human-subject study to elicit normative requirements
- Participants using Coco-based methodology
  - produce specifications with higher coverage and correctness;
  - expend equal time; and
  - feel Coco-based methodology is easy

Future directions

- Dealing with cycles in dominance
- Applying priority to a set of dominance relations

Thank you

Multiagent Systems and Service-Oriented Computing Laboratory, NCSU
http://research.csc.ncsu.edu/mas/