A Logic for Decidable Reasoning about Services

Yilan Gu
Dept. of Computer Science
University of Toronto

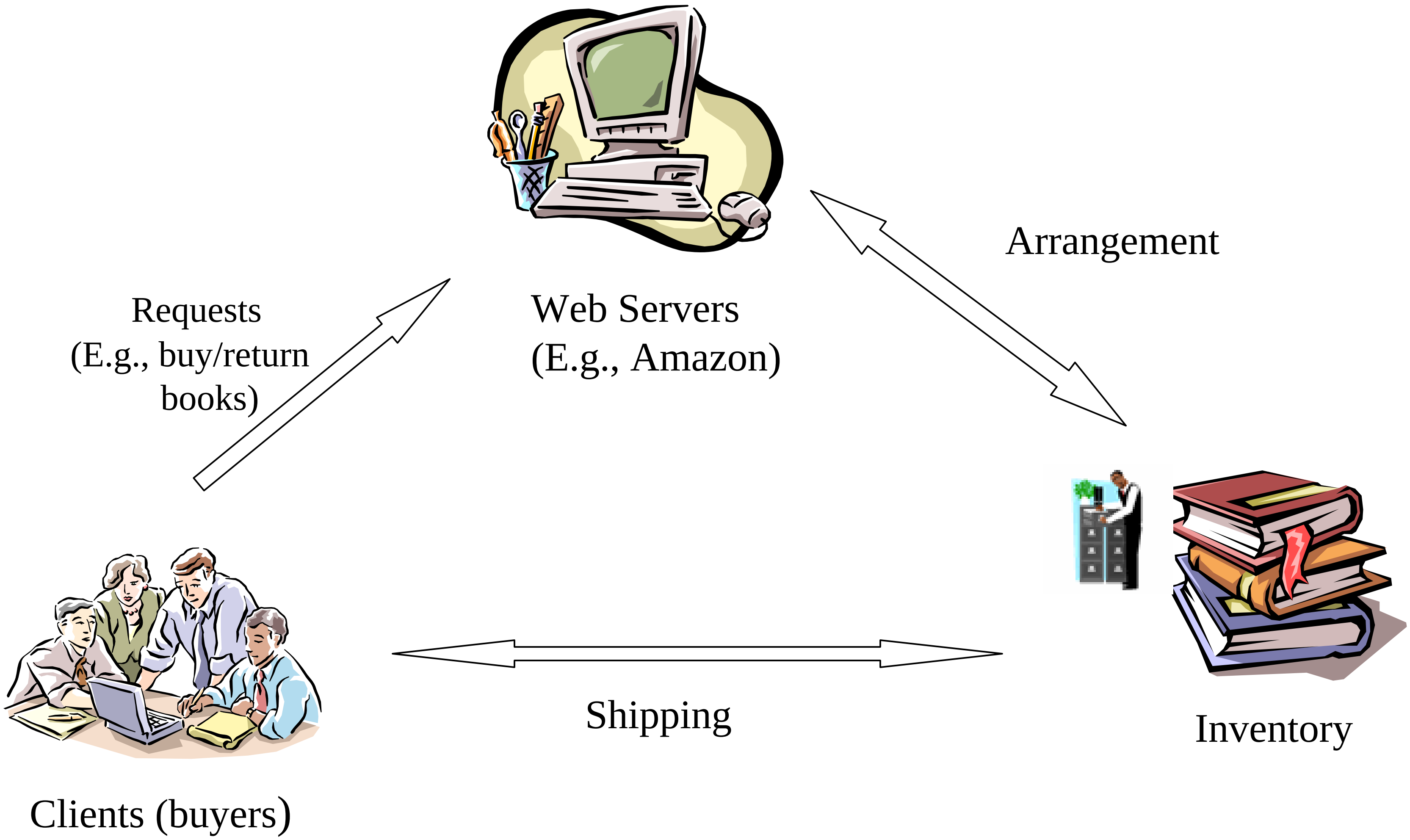
Mikhail Soutchanski
Dept. of Computer Science
Ryerson University

July 16, 2006

Outline

- Motivations
- Preliminaries
- Specification of the modified situation calculus for services
- Decidable reasoning about actions in this logic
- Discussions and future work

Shopping Online



Motivations

- Usually suppliers (Web servers) could not get complete information (OWA)
- Need composition of atomic services to achieve the clients' requests
- Integrating Semantic webs with Web services
- Representing the dynamics
  - What needs to be represented?

    atomic services (i.e., actions), dynamic environment (such as what books are available currently), the effect of service action
  - Expectations:
    - Represent actions for large/infinite domains (such as people, weight, time)
    - Be able to represent knowledge such as *"there exist some …"*
- For composite services and the environment,
  - What do we care about ? (reasoning)
    - Whether the composite services are possible to be executed successfully?
    - Whether certain properties/goals can be satisfied after the execution?
  - Expectations: efficient reasoning (here, decidability)

The Situation Calculus

- A first-order logic language
  - Represent actions and effects in a natural way
  - Very compact
- Three sorts:
  - Actions: *buyBook(x,y), returnBook(x,y),* …
  - Situations: S_0, *do*(*a*,*s*), *do*([a_1,…,a_n],*s*)
  - Objects: things other than actions and situations.

    E.g., places, names, numbers, etc.
- Fluents: system features whose truth values may vary.

  E.g., *instore*(*x*,*s*), *boughtBook*(*x*,*y*,*s*), *bought*(*x*,*y*,*s*)…

Basic action theory $\mathcal{D}$

- A set of first-order axioms to model actions and effects
- Precondition axioms for actions $\mathcal{D}_{ap}$:

  $Poss(buyBook(x,y),s) \equiv client(x) \wedge book(y) \wedge instore(y,s)$
- Successor state axioms $\mathcal{D}_{ss}$:

  $bought(x,y,do(a,s)) \equiv a = buybook(x,y) \vee a = buyCD(x,y)$

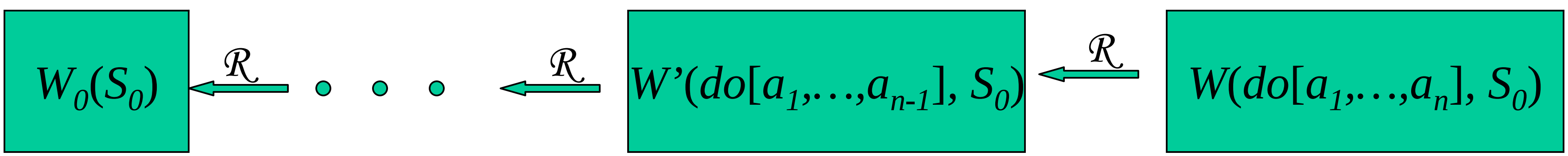
  $bought(x,y,s) \wedge \neg (a = returnbook(x,y) \vee a = returnCD(x,y))$
- Axioms for initial database $\mathcal{D}_{S0}$:
  - Knowledge known to be true in the situation S_0
  - Non-changeable facts
  - Open world assumption

Reasoning about Actions

- E.g., $(\exists x)(\forall y)(\forall y')\ boughtBook(x,y,S) \wedge boughtBook(x,y',S) \supset y=y'$
- Key reasoning mechanism -- regression operator $\mathcal{R}$
  - Successor state axioms support regression in a natural way

    If $F(x_1,\ldots,x_n,do(a,s)) \equiv \Psi_F(x_1,\ldots,x_n,a,s)$, then

    $$\mathcal{R}[F(t_1,\ldots,t_n,do(a',S))] = \mathcal{R}[\Psi_F'(t_1,\ldots,t_n,a',S)].$$



- Important properties for regression
  - $\mathcal{D} \models W \equiv \mathcal{R}[W]$
  - $\mathcal{D} \models W$ *iff* $\mathcal{D}_{S0} \cup \mathcal{D}_{una} \models \mathcal{R}[W]$

Disadvantages of the Situation Calculus

Advantage: representing actions and effects very compactly.

Disadvantage: reasoning for actions in general is undecidable under the open world assumption (OWA).

1. Can we get rid of the disadvantage?
2. Can we specify the Semantic Web features in a natural way?

Solution: Consider a fragment of first-order logic C^2.

Description Logics v.s. C^2

- Description logics
  - Base of OWL
  - Different varieties
  - $\mathcal{ALCQIO}$ ($\sqcap$, $\sqcup$, $\neg$, | , *id*)
- C^2 – a fragment of FOL
  - At most two variables *x*, *y*
  - No function symbols
  - Add counting quantifiers $\exists^{\geq n}$, $\exists^{\leq n}$

- $\mathcal{ALCQIO}$($\sqcap$, $\sqcup$, $\neg$, |, *id*) v.s. C^2
  - Concept names $\Leftrightarrow$ unary predicates *instore* ------- *instore*(*x*)
  - Role names $\Leftrightarrow$ binary predicates *boughtBook* ------ *boughtBook*(*x*,*y*)
  - E.g., $\exists^{\geq n} R.C \Leftrightarrow \exists^{\geq n} y.R(x,y) \wedge C(y)$, $\forall R.C \Leftrightarrow \forall y.R(x,y) \supset C(y)$

    $\neg C \Leftrightarrow \neg C(x)$, $C1 \sqcap C2 \Leftrightarrow C1(x) \wedge C2(x)$

Decidability of DLs and C^2

- [Borgida1996] $\mathcal{ALCQIO}(\sqcap, \sqcup, \neg, |, id)$ plus cross-product $\Leftrightarrow C^2$.
- We showed that: $C1 \times C2 = (R \sqcup \neg R)|_{C2} \sqcap ((R \sqcup \neg R)|_{C1})^-$.
- [Grädel et al., Pacholski et al. 1997] C^2 is decidable even under OWA.

$\mathcal{ALCQIO}(\sqcap, \sqcup, \neg, |, id) \Leftrightarrow C^2$, the translation algorithm is linear to the size of the given formula.

$\mathcal{ALCQIO}(\sqcap, \sqcup, \neg, |, id)$ is decidable even under OWA.

- Other advantages
  - The features in Semantic Webs can be easily represented in C^2 .
  - The reasoning in C^2 can also be easily translated into DLs.
  - May use current existing efficient DL reasoners for C^2 formulas.

The Decidable Situation Calculus $\mathcal{L}_{SC}^{DL}$

- Sorts:
  - Terms of *objects* are either variable *x*, variable *y*, or constants
  - Action functions have at most two arguments
  - Variable symbol *a* of sort *action* and symbol *s* of sort *situation* are the only additional variable symbols

- Fluents with either two or three arguments:
  - (Dynamic) concepts *instore*(*x*,*s*), ….
  - (Dynamic) roles *boughtBook*(*x*,*y*,*s*), *boughtCD*(*x*,*y*,*s*), *bought*(*x*,*y*,*s*), …

- Facts with either one or two arguments:
  - (Static) concepts *person*(*x*), *client*(*x*), *book*(*x*), *cd*(*x*),…
  - (Static) roles *hasCreditcard*(*x*,*y*), …

- Logic: add counting quantifiers $\exists^{\geq n}$, $\exists^{\leq n}$

The Basic Action Theory of $\mathcal{L}_{SC}^{DL}$

- Precondition axioms:
  - The RHS is C^2 if the situation argument *s* is suppressed
- Success state axioms:
  - Allow counting quantifiers
  - Variables *a* and *s* are free in the RHS of the axioms
  - Moreover, *x*,*y*, *a* and *s* are the only variables (both free and quantified)
- Axioms for initial databases: (with OWA)
  - Each axiom is C^2 if S_0 is suppressed

Purpose: to ensure the regression result is C^2 regardless S_0 .

Extensions of the Basic Action Theory

- Allowing specify certain features similar to DLs
- Acyclic TBox axioms:
  - Dynamic ones: $C(x,s) \equiv \Phi_C(x,s)$ (C – *defined* dynamic concept)
  - Static ones: $C(x) \equiv \Phi_C(x)$ (provided in the $\mathcal{D}_{S0}$)
  - The RHS is C^2 when the situation argument s is suppressed

  E.g., $valCust(x,s) \equiv person(x) \wedge (\exists^{\geq 3} y)\ bought(x,y,s)$

  $client(x) \equiv person(x) \wedge (\exists\ y)\ hasCreditcard(x,y)$

  - Reasoning: use lazy unfolding for Dynamic ones
- RBox axioms:
  - For taxonomic reasoning purpose
  - $R1 \supset R2$ for role $R1$, $R2$

  E.g. , $boughtBook(x,y,s) \supset bought(x,y,s)$, $boughtCD(x,y,s) \supset bought(x,y,s)$

  - Correctly compiled in $\mathcal{D}_{SS}$. I.e., $\mathcal{D} \models (\forall\ x,y,s).R1(x,y)[s] \supset R2(x,y)[s]$

Reasoning: Regression + Lazy Unfolding

- Expectations
  - Resulting formula should be C^2 if S_0 is suppressed
  - Be able to handle dynamic TBox axioms

- Reiter's regression operator is not suitable: introduce new variables

- Formula *W* that is regressable in $\mathcal{L}_{SC}^{DC}$
  - The situation term in *W* are ground
  - Variables in *W* can only include *x*, *y*

- Modified regression operator $\mathcal{R}$
  - When *W* is not atomic, the operator is still defined recursively

    E.g., $\mathcal{R}[W1 \wedge W2] = \mathcal{R}[W1] \wedge \mathcal{R}[W2]$, …
  - Add $\mathcal{R}[\exists^{\geq n} v.W] = \exists^{\geq n} v.\mathcal{R}[W]$
  - Reuse variables x and y when *W* is atomic (examples on the next slide)
  - When *W* is a defined dynamic concept, use TBox axioms (lazy unfolding)

A Regression Example in $\mathcal{L}_{SC}^{DL}$

$A1 =$ *buyCD*(Tom, BackStreetBoys),

$A2 =$ *buyBook*(Tom, HarryPotter),

$A3 =$ *buyBook*(Tom, TheFirm)

$\mathcal{R}[(\exists\, x).valCust(x, do([A1,A2,A3],S_0))]$

$= \mathcal{R}[(\exists\, x).\ person(x) \wedge (\exists^{\geq 3}\, y)\ bought(x, y, do([A1,A2,A3], S_0))]$ (lazy unfolding)

$= (\exists\, x).\ person(x) \wedge (\exists^{\geq 3}\, y)\ \mathcal{R}[bought(x, y, do([A1,A2,A3], S_0))]$

= … (recursively do regression using the successor state axioms)

$= (\exists\, x).\ person(x) \wedge (\exists^{\geq 3}\, y)\ [(x=\text{Tom} \wedge y = \text{TheFirm}) \vee$

$(x=\text{Tom} \wedge y = \text{HarryPotter}) \vee$

$(x=\text{Tom} \wedge y = \text{HarryPotter}) \vee$

$bought(x,y,S_0)\ ]$

Important Properties

Suppose W is a regressable formula of $\mathcal{L}_{SC}^{DL}$ with the basic action theory $\mathcal{D}$

- The regression $\mathcal{R}[W]$ terminates in a finite number of steps.
- $\mathcal{R}[W]$ is a C^2 formula if S_0 is suppressed
- $\mathcal{D} \models W \equiv \mathcal{R}[W]$
- $\mathcal{D} \models W$ *iff* $\mathcal{D}_{S0} \cup \mathcal{D}_{una} \models \mathcal{R}[W]$
- The problem whether is $\mathcal{D} \models W$ decidable
  - $\mathcal{D}_{S0} \cup \mathcal{D}_{una} \models \mathcal{R}[W]$ is a decidable reasoning in C^2 when S_0 is suppressed everywhere
- The executability problems and projection problems are decidable in $\mathcal{L}_{SC}^{DL}$

Discussions and Future Work

- Conclusions
  - Formalize a decidable language suitable for Web services
  - Have compact powerful expression power

- Other related researches
  - [McIlraith and Son 2002] assumes that sufficient information is available
  - [Berardi et al. 2003] uses propositional dynamic logic to model servicies

    e-services → constants, fluents → F(s) (propositional fragment of the situation calculus)
  - [Artale & Franconi 2001, Baader et al. 2003] extend DLs with temporal logics to capture the change of the world over time instead of caused by actions
  - [Baader et al. 2005] defines a service using a triple of sets of DL formulas

- Possible future work
  - Implementations
  - Consider the knowledge base progression/update problem in $\mathcal{L}_{SC}^{DL}$
  - Etc.