CTR-S: A Logic for Specifying Contracts in Semantic Web Services

Hasan Davulcu
Department of CSE
Arizona State University,
Tempe, AZ 85287-5406
hdavulcu@asu.edu

Michael Kifer, IV Ramakrishnan
Department of Computer Science
Stony Brook University
Stony Brook, NY 11794
{kifer, ram}@cs.stonybrook.edu
A Procurement Scenario

- I need a super-fast computer
- I can build it - but I need escrow
- Here is the escrow - I need to finance the rest
- We need to evaluate your application
- If you cancel I will keep the escrow
- If I pay then you must deliver
- We offer both insured or regular delivery
- If we approve delivery should be insured
Semantic Web Services Requirements

- Service Discovery
- Contracting of Services
- Process Modeling
- Service enactment

(http://daml.org/services/swsl/requirements/)
**CTR-S: Concurrent Transaction Logic for Services Contracting**

- Extension of classical FO-Logic
- Well defined model theoretic semantics
- Sound proof theory
- Primitives for process modeling
- Primitives for contracting
- Can model and reason with dynamic situations
**CIR-S: Process Modeling**

- **Sequential composition:** \( \times \)
- **Parallel composition:** \( | \)

- **Reasoner’s (eg, client) Disjunction:** \( \lor \)
  
  \[
  \text{buy} \leftarrow \text{pay}\_escrow \times (\text{finance} \lor \text{sell})
  \]

- **Opponent’s (eg, service) Conjunction:** \( \land \)
  
  \[
  \text{sell} \leftarrow \text{reserve} \times (\text{deliver} \lor \text{keep}\_escrow)
  \]
  
  \[
  \text{deliver} \leftarrow \text{insured} \lor \text{non-insured}
  \]

- **Reasoner’s (eg, client) Disjunction:** \( \lor \)
  
  \[
  \text{finance} \leftarrow (\text{approve} \times (\text{make}\_pay \lor \text{cancel})) \land (\text{reject} \times \text{cancel})
  \]

- **Opponent’s (eg, service) Conjunction:** \( \land \)
  
  \[
  \text{non-insured} \leftarrow \text{delivered} \land \text{lost}
  \]
  
  \[
  \text{insured} \leftarrow \text{delivered} \land (\text{lost} \times \text{satisfy})
  \]
**CTR-S**: Constraints in Contract Execution

- **\( *e \)**: event e should always eventually occur
- **\( *e \lor *f \)**: either always event e occurs or always event f occurs
- **\( * (e \lor f) \)**: always either event e or f or both should occur
- **\( *e \land *f \)**: event e and f must always both occur
- **\( * (e \rightarrow f) \)**: if event e occurs, then f must also occur
- **\( * (e \rightarrow \neg f) \)**: it is not possible for e and f to happen together
**CIR-S: Procurement Scenario**

\[
\text{BUY } \leftrightarrow \text{ PAY}\_\text{ESCROW} \times (\text{FINANCE} \lor \text{SELL}) \\
\text{SELL } \leftrightarrow \text{ RESERVE} \times (\text{DELIVER} \lor \text{KEEP}\_\text{ESCROW}) \\
\text{DELIVER } \leftrightarrow \text{ INSURED} \lor \text{NON-INSURED} \\
\text{FINANCE } \leftrightarrow (\text{APPROVE} \times (\text{MAKE}\_\text{PAY} \lor \text{CANCEL})) \land (\text{REJECT} \times \text{CANCEL}) \\
\text{NON-INSURED } \leftrightarrow \text{ DELIVERED} \land \text{LOST} \\
\text{INSURED } \leftrightarrow \text{ DELIVERED} \land (\text{LOST} \times \text{SATISFY}) \\
\text{BANK: } \ast (\text{APPROVE} \land \neg \text{CANCEL} \rightarrow \text{SATISFY}) \\
\text{SELLER: } \ast (\text{CANCEL} \rightarrow \text{KEEP}\_\text{ESCROW}) \\
\text{BUYER: } \ast (\text{MAKE}\_\text{PAY} \rightarrow \text{DELIVER})
\]

\[
\text{BUY } \land (\text{BANK } \land \text{SELLER} \land \text{BUYER})
\]
CTR-S: Enforcing Constraints

- How to execute $\text{BUY} \land \Phi$?

- A set of logical equivalence based rewrite rules for most classes of constraints [pods98] …

- except disjunctive primitive $\star(\Phi_1 \lor \Phi_2 \lor ... \lor \Phi_n)$ ones

- we resort to equivalence transformations defined with the help of parse trees of CTR-S …
CTR-S: Enforcing $\ast (\Phi_1 \rightarrow \Phi_2)$

- Basic Idea

$\Pi \Phi_1 \Phi_1 \Phi_2 \Phi_2 \Phi_1$

+ CYCLE REMOVAL

$O(d^N \times B)$
Related Formalisms

- Processes: Petri Nets, Process Logics
- Constraints: Temporal Logics

CTR-S is designed to model and reason with dynamic situations that arise in service contracting …
Questions ?

THANK YOU!