Model Checking Correctness Properties of Electronic Contracts

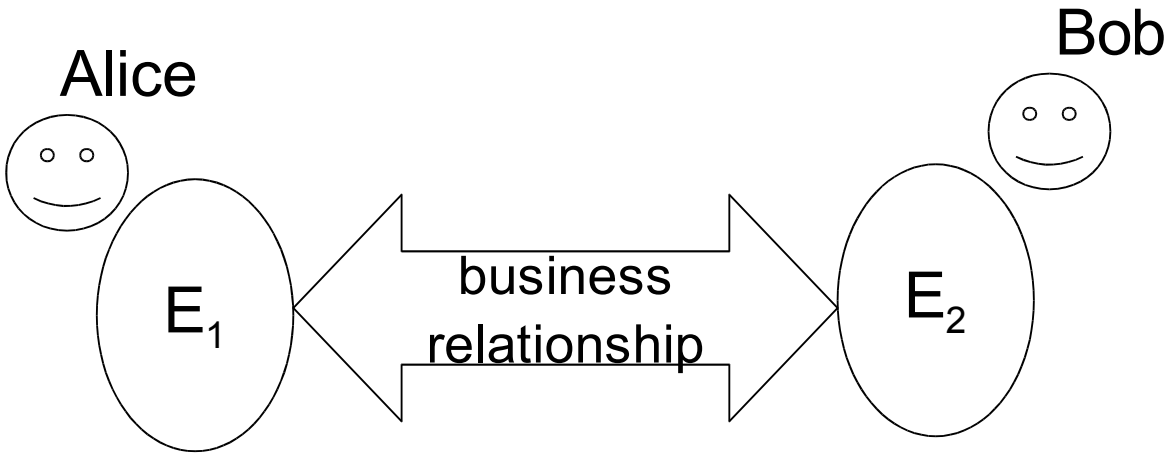
Carlos Molina-Jimenez
Carlos.Molina@ncl.ac.uk
Joint work with: Ellis Solaiman and Santosh Shrivastava
University of Newcastle upon Tyne

ICSOC 2003, Trento, 15-18 Dec, 2003

Outline of the presentation

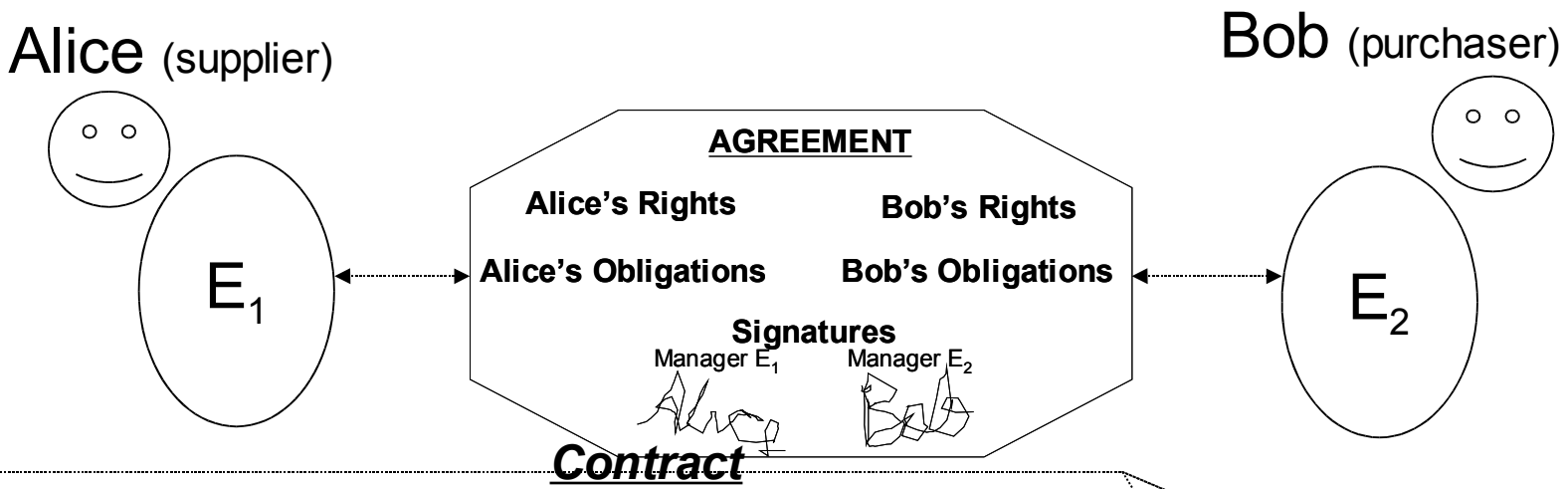
- Motivation for this presentation.
- Where are business contracts needed?.
- Representation of contracts with FSMs.
- Inconsistencies in contracts.
- Validation.
- Conclusions.

The general picture



- **Two or more autonomous business enterprises (E_1 , E_2).**
- **Wish to establish a business relationship.**
- **Cos' they're <u>mutually suspicious</u>, they need a <u>contract</u>.**

The general picture (cont.)



Contract

…

5. Offers

5.1 The supplier may use his discretion to send offers to the purchaser.

5.2 The purchaser is entitled to accept or reject the offer, but he shall notify his decision to the supplier.

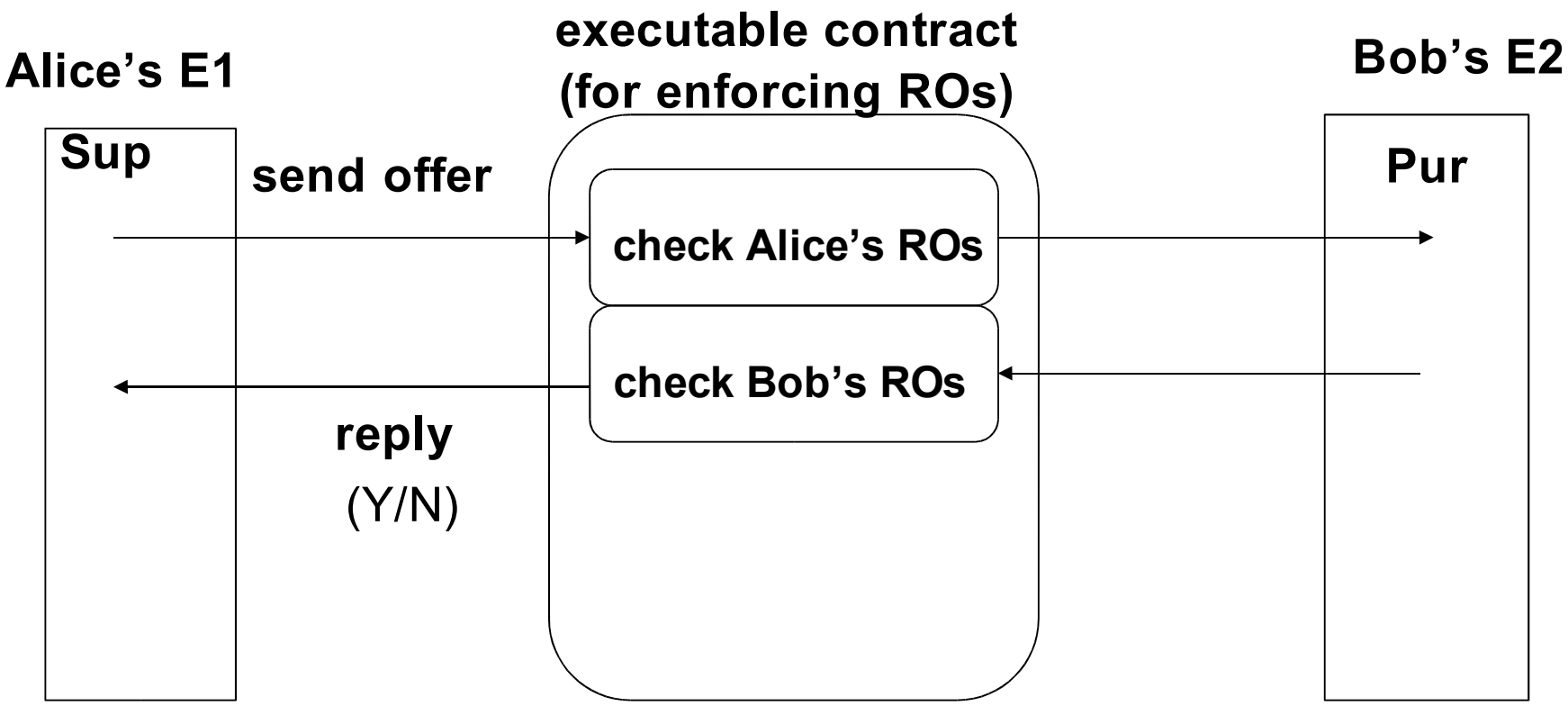
6. Commencement and completion

6.1 The contract shall start immediately upon signature.

…

We want executable contracts

Executable contract architecture



E-Enterprise, ROs- Rights and Obligations, Sup-supplier, Pur-purchaser

The problem

- Business contracts are full of logical inconsistencies.
  - "… the purchaser must send the payment to the supplier by 19 Dec 2003."
    - Contracts fail to specify:
      - What to do if the payment is incorrect.
      - How many times can the purchaser send incorrect payments.
      - That receiving of payments, purchase orders, complains, etc. must (must not) be acknowledged.
- Inconsistencies are meant to be detected, interpreted and corrected by humans (common sense).
- Computers don't have common sense.
  - Inconsistencies bring executable contracts into unexpected situations.

Contract validation process

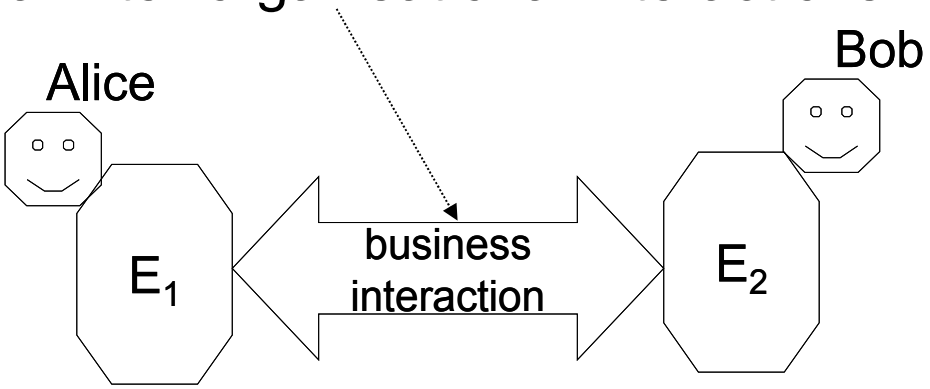
- Contract should be validated before implementation.
  1. Edit the English text contract.
  2. Convert the English text contract into a formal notation.
  3. Validate the formal notation.
     1. If inconsistencies were found go to 1.
  4. Implement contract.

Contract inconsistencies

- There are two sources of contract inconsistencies.
  1. Internal enterprise policies conflicting with contractual clauses.
     - The internal policies of $\mathbf{E_1}$ prohibit Bob from placing payments for chickens.
  2. Inconsistencies in the interaction between the business partners.
     - Alice is waiting for payment while Bob is waiting for item.
     - Alice is in end state while Bob is still running waiting for a confirmation.
- In this talk (paper), we address the second issue only.

Formal representation of contracts

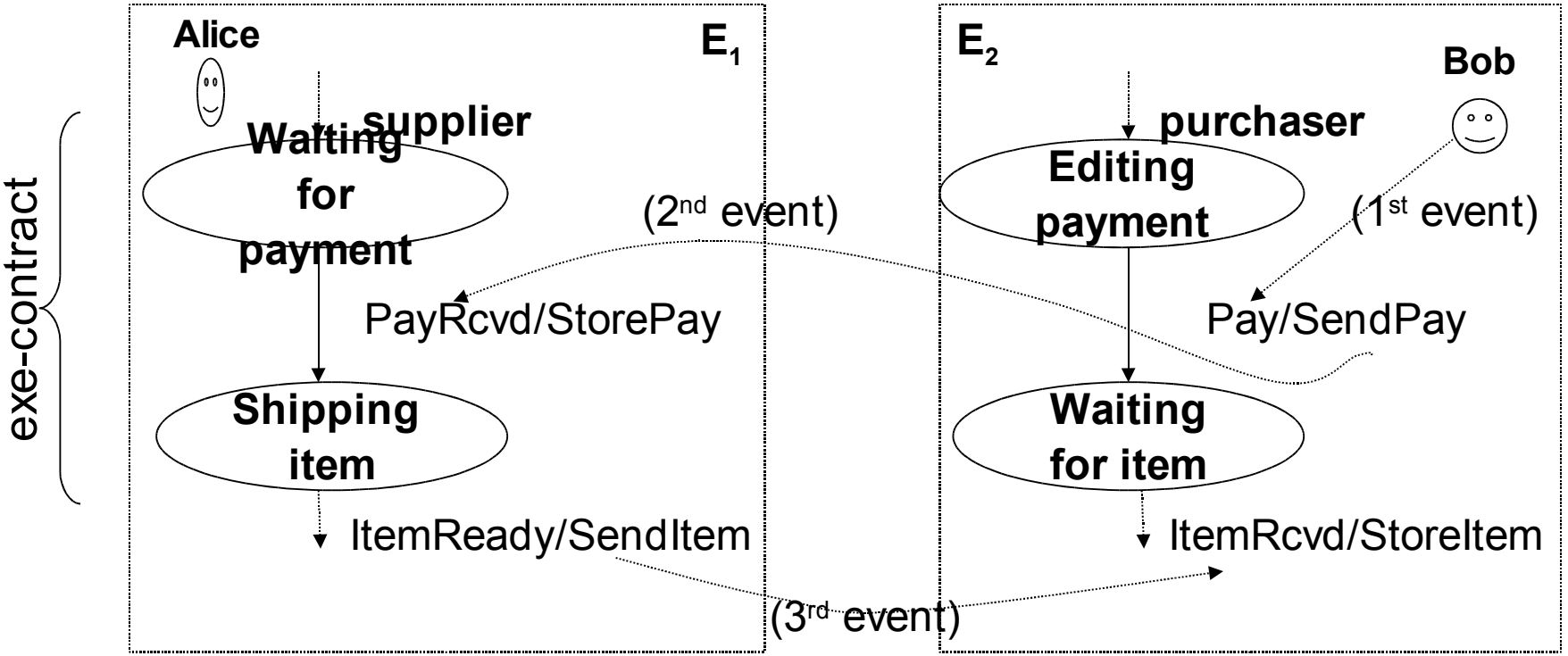
- There're several approaches.
- FSMs is (perhaps) the simplest one!.
- FSMs work PRETTY WELL for representing
  - Contractual inter-organisational interactions.



- Why do I like FSM-contracts?
  - Cos' they can be validated using standard model checkers.

A FSM-based executable contract

"The purchaser shall pay for the e-book before receiving it. The supplier shall send the e-book after receiving the payment."



legend: event/operation

Validation of correctness requirements

- The contracts are equivalent to communication protocols.
- They need to be validated to detect inconsistencies such as

  - Alice is waiting for the payment to arrive before shipping the item while Bob is waiting for the item.
    - **Deadlocks**
  - Alice is in end state while Bob is waiting for a confirmation.
    - **Incorrect termination**
  - Alice receives a second purchase order while she's expecting a payment.
    - **Unsolicited message**
  - Bob collected the item before paying for it.
    - **Precedence of operations**
  - Other correctness requirements.

Validation of correctness requirements with Spin

- In the paper we show how to use Spin for detecting this kind of inconsistencies in contracts.
- Spin is a mature and widely-used (in the academic community) model checker.
- It validates programs (eg contracts) written in Promela (modelling language).

Conclusions

- FSMs are powerful enough for representing contracts that control inter-organisational interactions.
- Contracts represented as FSMs can be easily validated using standard model checkers (e.g Spin).