Deontic logic for modelling data flow and use compliance

David Evans David M. Eyers

UNIVERSITY OF CAMBRIDGE

Motivation

To do useful work, ubiquitous systems operate on data

- The overall system should do more than the sum of its parts
- These data will be owned by different organisations
- These organisations will agree to contracts defining to data use

Our goals

- Encode data sharing policies in an intuitive way
- Check compliance against these data sharing policies
- When organisations are in conflict over policy compliance, the system should continue to function for others

Reasoning about state changes

- Participants move from one state to another
- Contradictions will be commonplace during contractual conflicts
- Classical predicate logic cannot represent modalities elegantly
- We need modalities to represent the state of compliance and thus make sense of these conflicts

Deontic logic

We focus on these deontic concepts:

Obligation Actions need to be performed to progress the state of affairs.

Prohibition An obligation not to perform some set of actions.

Violation An obligation not to do something is broken, or an obligation is left undone for too long.

Annulment The effect of some other predicate is cancelled out.

Event Calculus

Kowalski and Sergot's event calculus provides a straightforward mechanism for reasoning about changes in states of affairs.

Events are named, instantaneous effects that occur at some time

Fluents are named, half-open intervals in the same time domain

A set of application-specific rules relate fluents initiation and termination to event instances. Both events and fluents can be parameterised.

Event Calculus extensions

We extend the event calculus with the notion of fluents *deontically holding*.

- A fluent deontically holds if it holds, and it is not annulled by a fluent that deontically holds.

Contract representation

We want to preserve a direct correlation between source legal documents and the operation of the system

- We avoid explicit policy encodings that lead to workflow specifications
- We link deontic fluents to particular contract document sections
- The internal semantics of each document region might not be completely specified

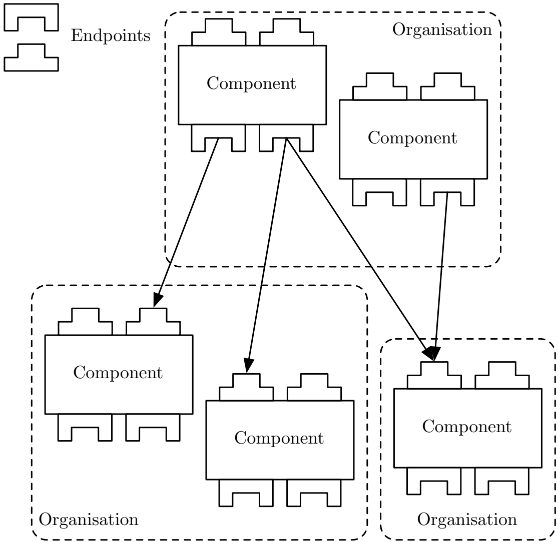
$\Rightarrow$ Triggers driven by or effects visible to humans ("red buttons" or "flashing lights") are natural

The TIME project

Our work in this area is focused on a project to construct a Transport Information Monitoring Environment (TIME)

- Collect data regarding the movements of pedestrians, buses, cyclists, vehicles, trains, *etc.*
- Monitor effects such as pollution, congestion, ...
- The data come from many different organisations

The TIME model



Contract labelling

Labelling involves construction of classes of organisations, data, and interactions.

1. Organisation classes are defined according to behaviour that is expected and useful, *e.g.*, “us”, “our partners who have signed an NDA”, and “our customers”
2. Types of data that might be exchanged are classified, *e.g.*, “proprietary data”, “data that we are willing to give to anyone”, *etc.*
3. Classes of interactions, and thus data exchange, are constructed.

The Deontic manager

Classes are used to construct events and fluents that correspond to the actions and states of compliance in the contract.

A special component—the *deontic manager*—is placed in each organisation to monitor the flow of messages passing in and out.

- Organisations and their deontic managers form mutual business relationships using out-of-band mechanisms.

Automatic monitoring

It may be appropriate to deploy software that triggers activation of deontic fluents automatically.

Fluents corresponding to the prohibition “information has been disclosed to a third party” may be able to operate by monitoring the communication network.

An appropriate user interface would map real-world, observable concerns to the events that are sent to appropriate deontic managers.

Implementation

Presently the deontic manager uses a Prolog-like implementation of the event calculus to monitor message transmissions

- The DM sends a message of its own when violation is suspected

In the future, the deontic manager might take proactive action such as discarding messages or effecting automatic conflict resolution

Conclusions

- Use as little abstraction as possible
- Closely couple implementation to requirements
- Provide a natural conduit for human interaction

Future work

- Performance measurement: ensuring that the deontic manager is not an unacceptable performance bottleneck
- Further contractual agreements
- Real system deployment