**Abstract**

*Institutions* provide an effective mechanism to govern agents in open distributed systems by specifying a set of *norms* (in terms of permissions, empowerments and obligations) regarding certain goals. However, when several institutions have to cooperate to govern the same entities simultaneously, *norm conflicts* are very likely to occur because institutions are typically designed independently with different goals. In this thesis, we aim to:

- identify the different ways to combine institutions.
- model those ways formally and computationally.
- detect conflicts in different combinations automatically.
- resolve those conflicts via automatic norm revision using an approach based on *inductive learning*.

**Cooperating Institutions**

![Diagram of three cooperating institutions](image)

**Three Forms of Cooperation**

(a) **Composite Institution**: a set of peer institutions combined together to form a common governance scope.
(b) **Multi-institution**: a hierarchical structure of interlinked institutions, in which an institution might be governed or influenced by other institutions.
(c) **Merged institution**: all the norms of each institutions are merged to form a completely new institution.

**Normative Conflicts**

- formalize institutions: event-driven action language *InstAL* [1].
- translate to computational model based on *Answer Set Programming*.
- automatic procedure to detect conflicting normative positions: *permission vs. prohibition, obligation vs. prohibition*.
- examine conflicts in all three combinations differently.

**Conflict Detection Mechanism**

Conflicts occur when two legal specifications disagree on a fluent known to both of them at the same time instant:

\[
\text{conflict}(\text{Inst}_X, \text{Inst}_Y, I, F):= \\
\quad \text{holdsat}(F \lor X, I), \text{not holdsat}(F \lor Y, I), \\
\quad \text{rename}(F, X, \text{Inst}_X), \text{rename}(F, Y, \text{Inst}_Y), \\
\quad \text{ifluent}(F), \text{ifluent}(Y), \\
\quad \text{instant}(I), \text{inst}(\text{Inst}_X; \text{Inst}_Y):
\]

**Conflict Resolution Mechanism**

- input conflicts and conflict traces.
- automatic norm revision via *Inductive Logic Programming (ILP)* [2].
- generate revision suggestions to resolve conflicts.
- select revision that minimizes the number of changes.

**Example**

**UK Immigration Law**: the permitted working hours of overseas students are reduced to 20 hours per week.

**StudentShip Regulations**: minimum number of teaching hours the student has to deliver is 40 hours per week.

**StudentShip Regulations**: initiated(obl(work(Student, 30), I)) :- \\
\quad \text{occurred}(\text{askForStudentShip}(\text{Student}, I)), \\
\quad \text{holdsat}(\text{availability}, I), \\
\quad \text{not holdsat}(\text{person}(\text{Student, overseas}), I).

- Conflicts arise for overseas students who are also granted by a studentship.
- Red part is suggested by the conflict resolution system.
- More details about this example can be found in [3].

**References**