Towards runtime support for norm change from a monitoring perspective

Motivation

Apply social abstractions to distributed systems in order to tame their complexity.

**Requirement:** Asses, at run-time the state of the normative Environment.

Regulative Norm

\[ \text{Min. Audient}_{i}(\text{angl}_j) \rightarrow \text{Charge}_{i}(\text{Prod}_j) \rightarrow \text{leave audient}_{i}(\text{angl}_j) \]

Constitutive Norm

Ride hand $\rightarrow$ Suggestion's Bid

Ride hand $\rightarrow$ Order Tour, Library Leave

Governance on Electronic Institutions

Basic Concepts:
- Language: $L$
- Ontology: $O$
- Logic connectives ($\land$, $\lor$, $\rightarrow$
- Set of all possible well-formed formulas: $\text{wff}(C_o)$ (DNF)
- A norm $n$ is a tuple $n = (l_0, s_0, l_1, s_1, \ldots)$
- A norm is considered fulfilled if, and only if:
  \[ l_0 - (Q_0, F_0, S_0) \leq (l_0, s_0) \]
- Event: $\langle (x, y, z) \rangle$
- Normative Monitor: $M_n = (N, S, IS, VS, FS, RS, E)$

Support for dynamic normative contexts

Non-static normative environments, evolving through time as regulations change adapting to new situations and behaviors.

Dynamic normative contexts, changing as new norms are added to the institution and removed from it. Under this conditions, it desirable to continue computing the state of the normative context at run-time, and computing states that are consistent with the modifications performed to the institution.

Retroactive Promulgation

Prospective Promulgation

Abrogation

Annulment

Norm Life Cycle

Conclusions

Formal generic method for expanding and contracting institutions at run-time

Formalisation of the four operations to be supported

Norm life-cycle extension

Algorithms

Ignasi Gómez-Sebastià, Sergio Alvarez-Napagao, Javier Vázquez-Salceda and Luis Oliva Felipe

* Universitat Politècnica de Catalunya, email: {igomez, salvarez, jvazquez, loliva}@lsi.upc.edu