What is a Norm?

A characterization of the normal

- Something accepted by many
- A standard
  - Formal or *de jure*
  - Informal or *de facto*

What is a Norm?

Emphasis in this course

- Some constraint or *rule of encounter* agreed to by the participants
- An elementary directed relationship between two parties

Must be violable!

Achieving Governance: Agents and Orgs

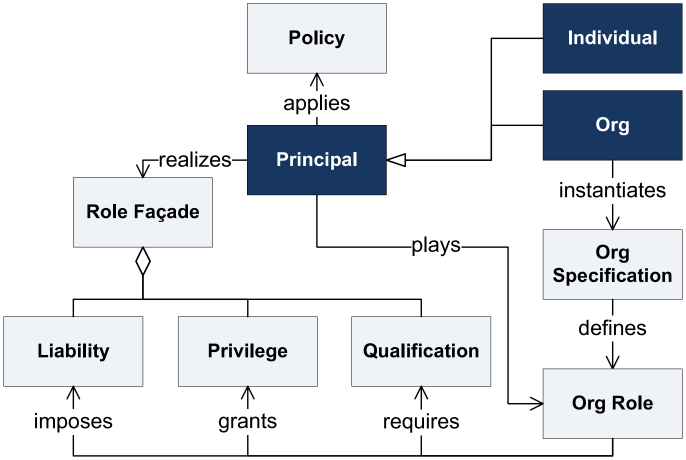
Put collaboration center stage

- Agents (including Orgs) are active computational entities, and represent principals
- Agents represent the stakeholders: people and organizations
  - Provide a locus for interaction
- Orgs are like *institutions:* have an identity and life time distinct from their members; also modeled as agents
  - Examples: NCSU, UNC System, ...
  - Provide a locus for roles and authorizations
  - Enforce behavioral constraints on members
    - Their main hold over their members is the threat of expulsion

Duality of Contracts and Orgs

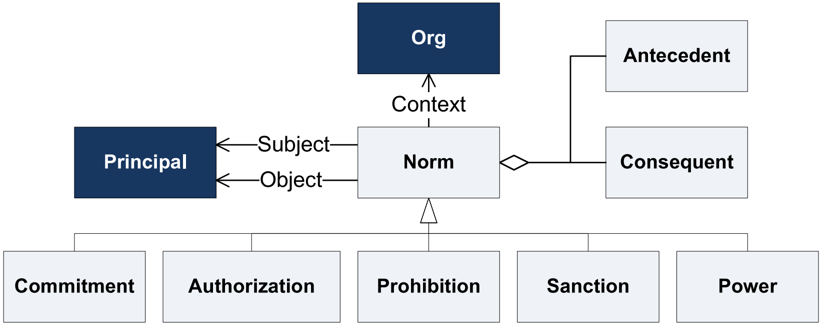
- A set of contracts define an Org
  - Roles, with their qualifications, privileges, liabilities
- An Org provides the context for defining contracts

Governance Conceptually



Additional Norms

Directionality and context are key features



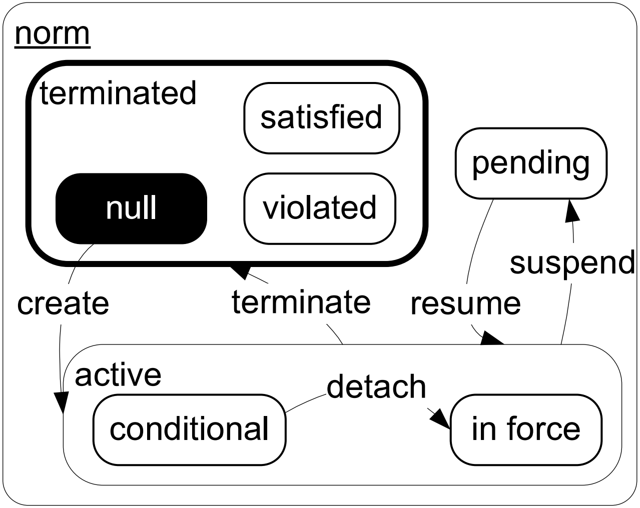
Permission versus prohibition and sanction

Norms and Façades

Read as: The subject is committed to the object . . .

Normative Concept	**Subject's Façade**	**Object's Façade**
Commitment	Liability	Privilege
Authorization	Privilege	Liability
Power	Privilege	Liability
Prohibition	Liability	Privilege
Sanction	Liability	Privilege

Norm Life Cycle: 1



Norm Life Cycle: 2

Computing the substate of a terminated norm

If terminated in		**Then**				
ant	**con**	**Com**	**Aut**	**Pro**	**San**	**Pow**
false	false	null	null	null	null	null
false	true	sat	vio	null	null	null
true	false	vio	null	sat	null	vio
true	true	sat	sat	vio	sat	sat

In the case of a power, a vio occurs upon the failure of an attempt to bring about the consequent.

Juggling Contract: 1

Contract For: Canterbury Centre Dinner 2003 ("CCD"), Friday 6 June 2003, 24 High Street, Canterbury

This agreement is entered into between the University Juggling Club ("UJC") and the Canterbury Center Dinner 2003 on the following terms:

1. Service Provider: University Juggling Club.
2. Employer: Canterbury Center Dinner.
3. To be provided by UJC: Performers: J Woods (juggler); one other juggler; all equipment necessary for performance.
4. To be provided by CCD: Cloakroom.
5. Venue address: 24 High Street, Canterbury.
6. CCD understands that performances are restricted in venues with ceilings of insufficient height. The ideal height is 5 meters. Outside performances are restricted in rain or strong winds.
7. Date of Performance: Friday 6 June 2003, starting at 6:30PM.
8. Duration of Performance: 1.5 hours. Short (less than one minute) breaks are part of the performance.

Juggling Contract: 2

9 Fee: £30 per juggler + £10 expenses + £90 insurance (total £160).

10 If UJC is forced to cancel, all monies (including £90 deposit) will be refunded in full. If the Employer cancels with at least 24 hours notice, UJC will retain £90 and return any other monies.

11 Should poor weather mean that the Event takes place indoors, UJC will refund £10 expenses.

12 Performers will not consume any alcohol until after completion of services as agreed.

13 CCD will be responsible for compensation to UJC for damage to equipment caused by those attending the Event unless damage is caused when (if) Performers have left equipment unguarded.

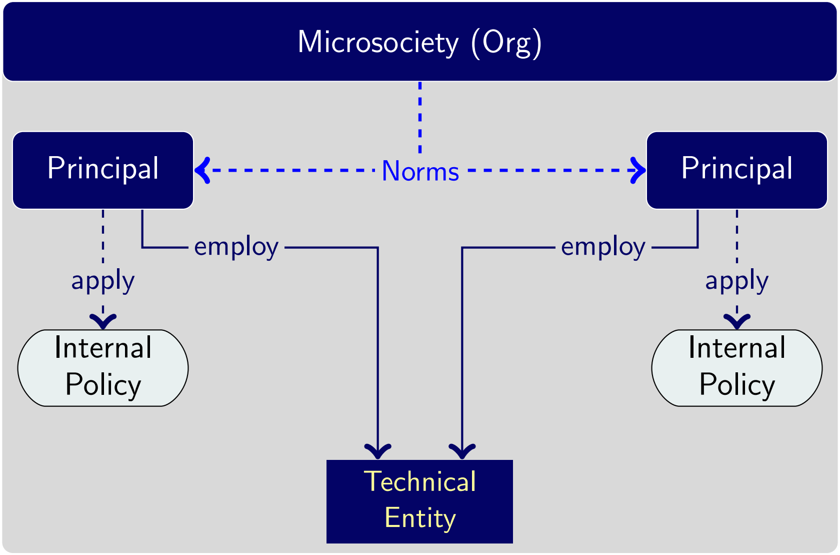
14 UJC will be liable for any injury sustained by a guest at the Event if such injury results from provision of services as agreed upon in this contract unless the Event fails to provide a suitable area for performance.

What is a Sociotechnical System?

A microsociety

- Social entities, i.e.,
    - Autonomous principals
- Interacting with one another
    - Through and about technical entities
- *Norms*: Standards of correct behavior with which to judge self and others
- Must respect autonomy: a principal may apply any policy to decide its actions
    - May violate norms

Simple Normative Framework for Sociotechnical Systems

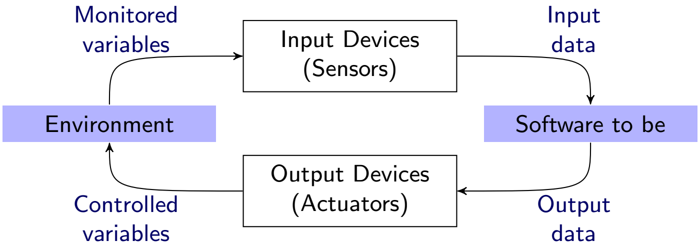


Sociotechnical Systems in Current Requirements Engineering

- An STS
  - Characterizes the organizational context of software
  - Represents the problem world, which must be analyzed to derive requirements specifications for software
- In current RE,
  - Autonomy and norms are deemphasized
  - Devices and legacy software are modeled as "actors" on par with social entities

Traditional Requirements Engineering: Machine-Oriented

RE considers only two “participants”: machine (software) and environment



- Methodologically: Refine requirements into a machine specification
- We classify approaches such as Tropos and KAOS as traditional
  - Even though they consider many actors (“agents”), which enables fine-grained analysis of the sociotechnical environment

Specifying Sociotechnical Systems

Why restrict ourselves to software (the "technical entity")?

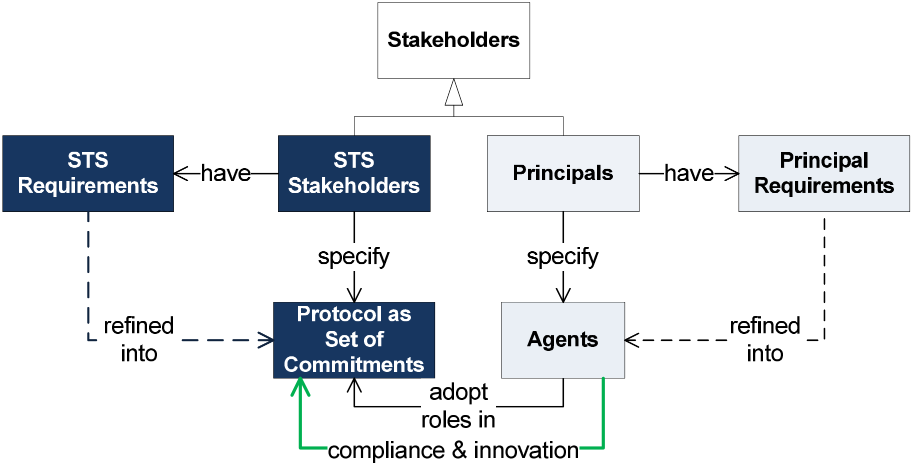
- What is the nature of a specification for STSs?
- How may we derive such a specification from requirements?
- How do STS specifications relate to machine specifications?
- What do we gain from specifying an STS and then the machines instead of specifying the machines directly?

Protocol: Specification of Commitments

Commitment is the kind of norm supported in Protos

- C(Buyer, Seller, goods, pay): Active and conditional
- If *goods* $\wedge$ C(Buyer, Seller, goods, pay) Then
  - Active and detached (or unconditional or base)
  - C(Buyer, Seller, T, pay)
- If C(Buyer, Seller, T, pay) Then
  - If *pay* Then Satisfied
  - If never *pay* Then Violated
- If C(Buyer, Seller, goods, pay) Then
  - If *pay* Then Satisfied
  - If never *pay* and never *goods* Then Expired

Protos Stakeholders and their Artifacts



Regimentation: Preventing Violations of Norms

- In an automobile insurance STS, suppose we have this requirement:
  - Any replacement parts ordered by a mechanic that are priced above $500 must be approved by the insurer
- Traditional approaches rely on technical solutions: produce machine specifications to enforce some requirement
  - Specify a machine that disables orders for parts over $500 unless the insurer approves them first
- Regimented solutions curtail deviations and innovations
  - What happens if a mechanic wants to urgently place the order?

Regulation: Stating a Norm but Allowing its Violation

Previous requirement as a norm

- $\{\mathsf{C}(\textsc{Mechanic}, \textsc{Insurer}, \mathsf{price}(\textit{item}) > 500, \mathsf{getApproval}(\textit{item})$ precedes $\mathsf{order}(\textit{item}))\}$
- Benefits
  - Captures the essence of the requirement
  - Enables determining accountability and compliance
  - Enables a principal to design agent according to own requirements, perhaps one that is designed to be noncompliant in some situations

Characterizing Requirements Engineering

- Machine orientation (Zave and Jackson)

$$A, M \vdash R$$

- Sociotechnical orientation

$$A, Ag, S_P \vdash R$$

  - Collapses to Z&J when Ag is a singleton and, therefore, S_P is a null protocol
  - Principals via their agents are runtime participants in the system
  - An STS is a multiagent system, not just an environment and machine

Social Refinement

Deriving protocol specifications from requirements

- Given an initial configuration $\langle S, A, N \rangle$
  - N, a finite set of requirements (needs)
    - For example, $\{$R$(\textsc{c}, \textit{prepared})$, R$(\textsc{i}, \textit{sold})$, R$(\textsc{m}, \textit{paid})\}$
  - A, an empty set of assumptions
  - S, an empty set of assumptions
  - Apply a reduction rule to produce a new configuration $\langle S', A', N' \rangle$
- Design process: successive application of the rules
  - Consideration of alternative rules
  - Would need to be made concrete

Need Refinement (Traditional in Software Engineering)

If p refines to p', a need for p can be met by meeting a need for p'; record assumption that p refines to p'

$$\langle S, A, N \cup \{\mathsf{R}(\tau, p)\}\rangle \hookrightarrow \langle S, A \cup \{p \hookrightarrow p'\}, N \setminus \{\mathsf{R}(\tau, p)\} \cup \{\mathsf{R}(\tau, p')\}\rangle$$

	S	**A**	**N**
1	$\varnothing$	$\varnothing$	$\mathsf{R}(\textsc{c}, \mathit{prepared})$, $\mathsf{R}(\textsc{i}, \mathit{sold})$, $\mathsf{R}(\textsc{m}, \mathit{paid})$
2	$\varnothing$	$\{\mathit{prepared} \hookrightarrow \mathit{covered} \wedge \mathit{eme}\}$	$\mathsf{R}(\textsc{c}, \mathit{covered} \wedge \mathit{eme})$, $\mathsf{R}(\textsc{i}, \mathit{sold})$, $\mathsf{R}(\textsc{m}, \mathit{paid})$

Commitment Introduction (New for Sociotechnical Systems)

If τ_1 has a need for q, τ_1 can address that need by obtaining a commitment from τ_0 to τ_1 whereby τ_0 commits to bringing about q provided p holds; τ_1 takes on the need for p

$$\langle S, A, N \cup \{\mathsf{R}(\tau_1, q)\}\rangle \hookrightarrow \langle S \cup \{\mathsf{C}(\tau_0, \tau_1, p, q)\}, A, N \setminus \{\mathsf{R}(\tau_1, q)\} \cup \{\mathsf{R}(\tau_1, p)\}\rangle$$

	S	**A**	**N**
2	$\varnothing$	$\{\mathit{prepared} \hookrightarrow \mathit{covered} \wedge \mathit{eme}\}$	$\mathsf{R}(\textsc{c}, \mathit{covered} \wedge \mathit{eme})$, $\mathsf{R}(\textsc{i}, \mathit{sold})$, $\mathsf{R}(\textsc{m}, \mathit{paid})$
3	$\mathsf{C}(\textsc{i}, \textsc{c}, \mathit{sold}, \mathit{covered})$	$\{\mathit{prepared} \hookrightarrow \mathit{covered} \wedge \mathit{eme}\}$	$\mathsf{R}(\textsc{c}, \mathit{sold} \wedge \mathit{eme})$, $\mathsf{R}(\textsc{m}, \mathit{paid})$

Design Logic and Soundness

- A set of useful reduction rules
  - Cyclic commitments, subteam, and so on
- A design path for requirement R is a sequence of configuration $\langle S_0, A_0, N_0 \rangle, \ldots, \langle S_n, A_n, N_n \rangle$ such that
  - $\langle S_0, A_0, N_0 \rangle$ is initial for R
  - $\langle S_n, A_n, N_n \rangle$ is final for R, that is, N_n is empty
- The design path is sound if $\langle S_n, A_n, N_n \rangle$ is consistent and satisfies R
  - Notice that because N_n is empty, effectively, S_n and A_n satisfy R
- We specify a logic and show that design paths produced by applying the rules are sound

Exploring the Design Space

Applying different reductions yields alternatives

$$\begin{tikzcd}
 & \mathrm{R}(y,q) \arrow[d, dash, "\text{Begin design}"] & \\
 & S=\{\};\ A=\{\};\ N=\{\mathrm{R}(y,q)\} \arrow[dl, dash, "\text{Introduce commitment}"'] \arrow[dr, dash, "\text{Do it in-house}"] & \\
S=\{\mathrm{C}(x,y,p,q)\};\ A=\{\};\ N=\{\mathrm{R}(y,p)\} \arrow[d, dash, "\text{Accept onus}"'] & & S=\{\};\ A=\{\mathrm{O}(y,q)\};\ N=\{\} \\
S=\{\mathrm{C}(x,y,p,q)\};\ A=\{\mathrm{O}(y,p)\};\ N=\{\} & &
\end{tikzcd}$$

Software Engineering for Sociotechnical Systems

- Idea that STS specifications are not machine ("software") specifications but protocol specifications
  - Expanded scope of software engineering
  - Specification of protocols is independent from the specification of principals' agents
    - Enables principals to act flexibly and in innovative ways
- Social refinement
  - Reduction rules for deriving protocol specifications from STS stakeholder requirements
  - Logic for relating satisfaction of requirements to protocol specifications and assumptions

Directions

Protos: An anagram of Tropos

- Handling conflicting requirements, potentially by satisficing rather than satisfying requirements
- Extensions to other kinds of norms, e.g., authorization, power
- Formalizing other useful reductions, e.g., for capturing delegation
- Devising methodologies and tools
  - To give guidance to STS stakeholders on how to apply the reductions
  - To give guidance to agent designers