Coco-M: Designing Sociotechnical Systems (Group A)

Healthcare Scenario

Whenever Alice takes ill, Bob her parent takes her to a hospital. Bob also works under Steve, his employer, who expects him to work during work hours. One day, Alice fell sick during Bob's work hours. Bob disregards his employer's expectation in light of the medical emergency involving his ward.

Background

Finite State Machine

To model the behavior of a system, we use *state machines*. State machines indicate different states of a system at different times. The *state* of a system is considered as behavior of a system at any given time. A system can change from one state to another by triggering an action or an event. We call this action or event as a *transition*. Examples of state machines include vending machines, elevators, traffic lights, combination locks, and so on.

A finite state machine has the following properties

- The system must have finite number of states (S)
- The system has a particular initial state (s_0)
- The system must have finite number of inputs or events that can trigger transitions between states ($\sum$)
- The behavior of a system at the given time depends on the current state and the input or event that occur at that time
- The system has a set of final states (F)
- The state transition function is δ: S $\times$ $\sum$ $\rightarrow$ S

Consider an example of a turnstile that has two states: S = $\langle$Locked, Unlocked$\rangle$. There are two inputs: $\sum$ = $\langle$coin, push$\rangle$. Coin indicates putting a coin in the slot. Push indicates pushing the arm. Consider the initial state as the locked state. In the locked state, the arm of the turnstile cannot be moved irrespective of how many times the arm is pushed (δ: Locked $\times$ push $\rightarrow$ Locked). However, when a coin is put, the state of machine moves from the locked state to the unlocked state (δ: Locked $\times$ coin $\rightarrow$ Unlocked). Putting an additional coin doesn't change the state (δ: Unlocked $\times$ coin $\rightarrow$ Unlocked). Now, when the arm is pushed, the state again moves from the unlocked state to the locked state (δ: Unlocked $\times$ push $\rightarrow$ Locked). Figure 1 represents the state diagram for a turnstile with its states, inputs, and state transitions.

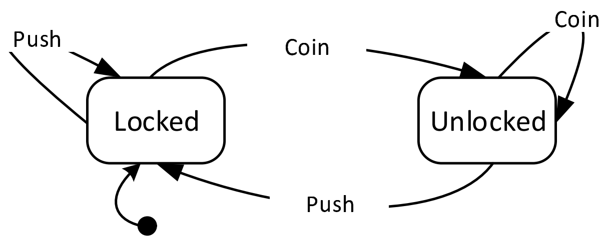


Figure 1: State diagram for a turnstile.

Norms

Humans, organizations, and technical systems such as software interplay with each other in a *sociotechnical* system. To capture the requirements of sociotechnical systems, we adopt Singh's [1] model of norms. A norm is directed from a subject to an object and is constructed as a conditional relationship involving an antecedent (which brings the norm in force) and a consequent (which brings the norm to satisfaction). This representation yields clarity on who is accountable to whom. A norm has four core elements—SUBJECT, OBJECT, antecedent, and consequent. It can be formalized as $\mathsf{N}(\text{SUBJECT}, \text{OBJECT}, \textit{antecedent}, \textit{consequent})$. Here, SUBJECT and OBJECT are roles where as antecedent and consequent are disjuction or conjunction of actions and events.

A commitment means that its *subject commits to its object* to ensure the consequent if the antecedent holds. For example, in a hospital, physicians are committed to the hospital to operating upon patients when there is an emergency. We write a commitment as:

$\mathsf{C}(\text{PHYSICIAN}, \text{HOSPITAL}, \textit{emergency}, \textit{operate})$

An authorization means that its *subject is authorized by its object* for bringing about the consequent if the antecedent holds. For example, physicians are authorized by the patients to access their electronic health record (EHR) if the patients give consent.

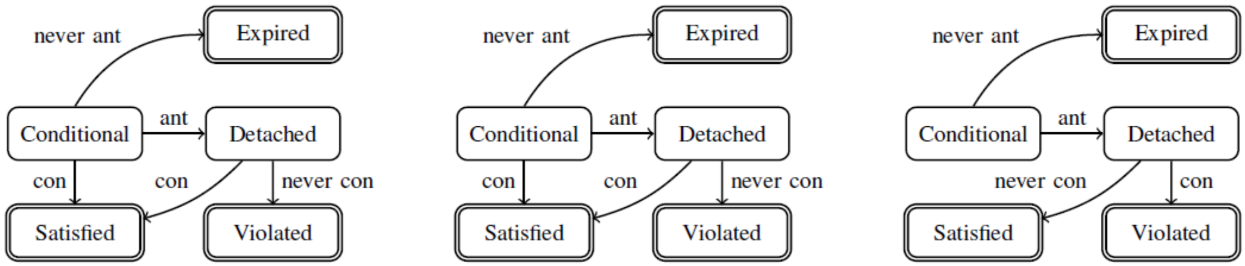
We write a authorization as:

$\mathsf{A}(\text{PHYSICIAN}, \text{PATIENT}, \textit{consent}, \textit{accessEHR})$

A prohibition means that its *subject is forbidden by its object* from bringing about the consequent if the antecedent holds. For example, health care professionals (HCP) are prohibited by the hospital from disclosing patients protected health information (PHI). We write a prohibition as:

$\mathsf{P}(\text{HCP}, \text{HOSPITAL}, \textit{true}, \textit{disclosePHI})$

Norm Lifecycle



(a) Commitment (accountable: subject) (b) Authorization (accountable: object) (c) Prohibition (accountable: subject)

Figure 2: The lifecycle of commitment, authorization, and prohibition norms.

Figure 2 represents the lifecycle of commitment, authorization, and prohibition norms. The lifecycle for each commitment is represented as a state machine with finite number of states S = $\langle$conditional, detached, satisfied, violated$\rangle$, a set of inputs $\sum = \langle$never ant, ant, con, never con$\rangle$, initial state s_0 = conditional, and various state transitions δ: S $\times \sum \rightarrow$ S. In the input, ant represents antecedent and con represents consequent. Consider the example of the commitment norm, C(PHYSICIAN, HOSPITAL, emergency, operate) where PHSYCIAN commits to HOSPITAL to operate a patient in case of emergency. Intially, the commitment is in the conditional state. Once, the ant or emergency occurs, the norm moves from the conditional state to the detached state. When PHSYICIAN operates, the norm moves from the detached state to the satisfied state. If PHYSICIAN doesn't operate, the norm moves from the detached state to the violated state. If the norm is in the conditional state, and emergency never occurs, the norms moves to the expired state.

In Figure 2, we can observe that the state transitions vary from one norm to another. This is due to the specific property of each norm. For example, in the commitment and the authorization norm, if the norm is in the detached state, a con input moves the norm from the detached state to the satisfied state. Compared to the commitment and the authorization norm, if the prohibition norm is in the detached state, a never con input moves it from the detached state to the satisfied state.

Coco: Approach Outline

Defining Normative Model

- Identify subscenarios in the given scenario, e.g., "Bob takes Alice to a hospital when Alice is sick".

- Identify all actors involved in a subscenario, e.g., *Alice*, *Bob*, *Hospital*, and *Steve*.

- Develop abstract actors as roles where appropriate, e.g., *child*, *guardian*, *employer*, and *employee*:

- Identify all actions in a scenario, as in "Bob takes Alice to the hospital when Alice is sick" and "Bob works for Steve during work hours"
- Develop abstract actions from the identified actions, as in "Guardian ensures child is treated when sick" and "Employee works for employer during work hours"
- Identify contexts in the actions, as in "Child is sick" and "work hour"
- Think of relevant norms, as in "Guardian is committed to the child to get it treated", and associate context with norms:
  - gCom: In a society, guardian is committed to his or her child to take him or her to a hospital when the child is sick.

    C_{gCom}: $\mathsf{C}(\text{GUARDIAN}, \text{CHILD}, sick, bringToHospital)$
  - wCom: In a company, an employee is committed to his or her employer to work during work hours.

    C_{wCom}: $\mathsf{C}(\text{EMPLOYEE}, \text{EMPLOYER}, workhour, work)$
- Identify possible conflicts and inconsistencies, e.g., "A working guardian (guardian who is also an employee) cannot take his or her child to a hospital and go to work at the same time."
- Explain how the conflict is reached, if there is any reason, e.g., "A guardian or an employee cannot be at two places at the same time, assuming the hospital and the company he or she works for are at different locations."
- Resolve conflicts by capturing contextual preference between norms, e.g., "gCom is preferred to wCom."

Designing State Machines

From the normative model,

- Identify possible states, inputs, and transitions from the actions associated with each role and norms
- From identified states, inputs and transitions create state machines for each subscenario
- Create a final state machine by combining all state machines of individual subscenarios

Solution

Solution: Normative Model

The subscenarios in the healthcare scenario are

- Bob takes Alice to a hospital, when Alice is sick

- Bob works for Steve during work hours
- Bob disregards Steve's expectation by taking Alice to a hospital during work hours

The actors involved in the scenario are Bob, Alice, and Steve. In the first scenario, we can abstract Bob as GUARDIAN and Alice as CHILD. In the second scenario, we can abstract Bob as EMPLOYER and Steve as EMPLOYEE. In the third scenario, the roles are same as the first scenario.

We can identify two norms from the first two scenarios. In the first scenario, we can identify a commitment norm as C_{gCom}: C(GUARDIAN, CHILD, sick, bringToHospital). Here, the subject is GUARDIAN and the object is CHILD. The ant is *sick* and the con is *bringToHospital*. In the second scenario, we can identify a commitment norm as C_{wCom}: C(EMPLOYEE, EMPLOYER, workhour, work). Here, the subject is EMPLOYEE, EMPLOYER. The ant is *workhour* and the con is *work*. The third scenario indicates a set of specific events instead of any specific norm. Associating context to norms, C_{gCom} indicates a *social* context where as C_{wCom} indicates a *work hour* context.

In the third scenario, there is a contextual conflict. Considering the *social* context, GUARDIAN is expected to satisfy its commitment toward *Child*, whereas considering the *work hour* context, EMPLOYER is expected to satisfy its commitment toward *Employer*. To resolve the conflict, we can specify a preference. For example, we can state that in this specific case of conflict, C_{gCom} in the *social* context shall be preferred over C_{wCom} in the *work hour* context.

Figure 3 shows the identified roles, and norms.

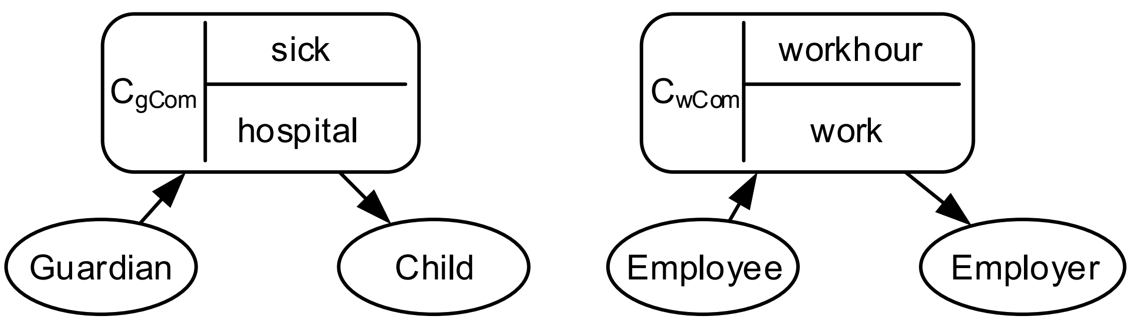


Figure 3: Roles and Norms.

Solution: State Machines

For each subscenario, we can come up with the following state machines

- The state machine from the first subscenario has the following: S = $\langle S_1, S_2, S_3\rangle$, $s_0 = S_1$, $\sum = \langle$sick, hospital$\rangle$, δ: $S_1 \times$ sick $\rightarrow S_2$, $S_2 \times$ bringToHospital $\rightarrow S_3$. S_1 indicates C_{gCom} is conditional, S_2 indicates that C_{gCom} is detached, and S_3 indicates C_{gCom} is satisfied.

- The state machine from the second subscenario has the following: S = ⟨S_1, S_4, S_5⟩, s_0 = S_1, $\sum$ = ⟨workhour, work⟩, δ: $S_1 \times$ work hour $\rightarrow S_4$, $S_4 \times$ work $\rightarrow S_5$. S_1 indicates C_{gCom} is conditional, S_2 indicates that C_{gCom} is detached, and S_3 indicates C_{gCom} is satisfied.

- The state machine from the third subscenario has the following: S = ⟨S_1, S_6, S_7⟩, s_0 = S_1, $\sum$ = ⟨work hour $\wedge$ sick, hospital⟩, δ: $S_1 \times$ sick $\wedge$ work hour $\rightarrow S_6$, $S_6 \times$ pref(hospital) $\rightarrow S_7$. S_1 indicates C_{gCom} and C_{wCom} are conditional, S_6 indicates C_{gCom} and C_{wCom} are detached, S_7 are C_{gCom} and C_{wCom} are satisfied. In this case, the contextual preference *pref(hospital)* results in satisfaction of norms C_{gCom} and C_{wCom}. Without the presence of contextual preference, C_{gCom} would have moved to the satisfied state whereas C_{wCom} would have moved to the violated state.

Figure 4 represents the overall state machine for the scenario. In the state machine we can observe that there are two additional transitions δ: $S_2 \times$ workhour $\rightarrow S_4$ and $S_3 \times$ sick $\rightarrow S_4$. These additional transitions represent possible transitions in the combined state machines from each subscenario.

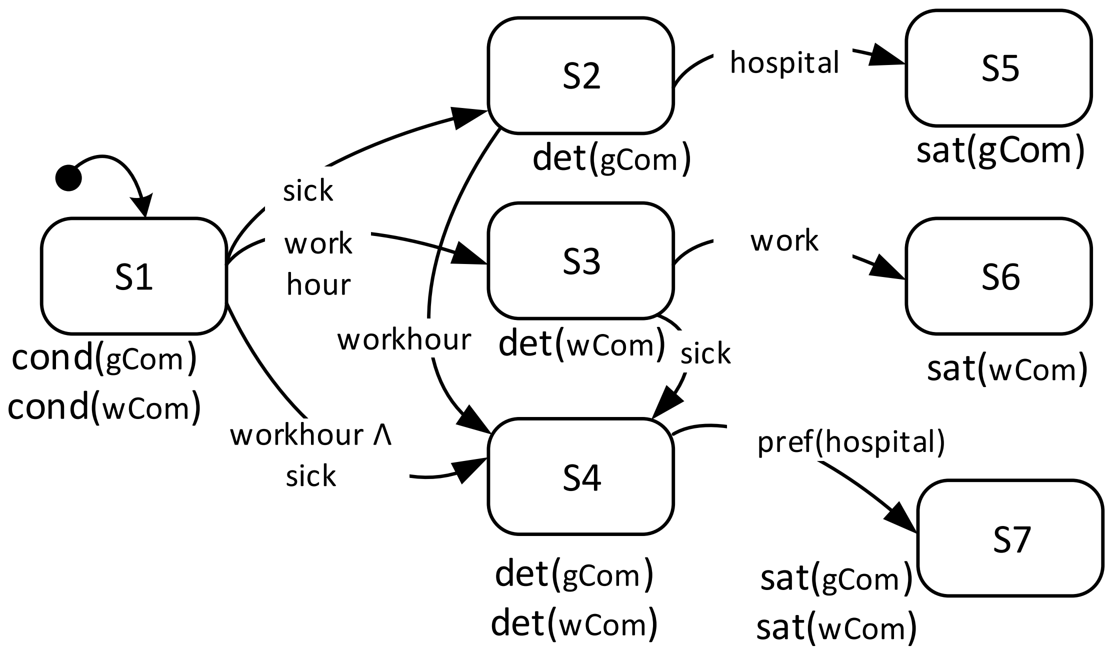


Figure 4: State machine for the healthcare scenario.

References

[1] Munindar P. Singh. Norms as a basis for governing sociotechnical systems. *ACM Trans. Intell. Syst. Technol.*, 5(1):21:1–21:23, December 2013.