

Engineering Adaptive Software Systems (EASSy)



Human Factors in Self-Adaptive Socio-Technical Systems

Xin Peng

**School of Computer Science
Fudan University**

pengxin@fudan.edu.cn

<http://www.se.fudan.edu.cn/pengxin>

Some of the works are done jointly with Yijun Yu and John Mylopoulos.



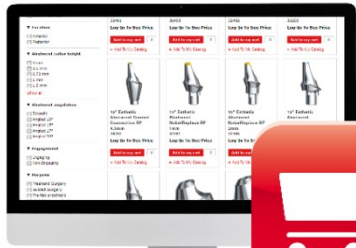


Socio-Technical Systems (STS)

- **Elements:** consist of human, organization, hardware and software components that work together to fulfill stakeholder requirements
- **Uncertainty:** operate under uncertainty as components fail, humans act in unpredictable ways, and the environment of the system changes
- **Multi-Agent:** each agent intends to best achieve its own goals relying on its own capabilities and other agents (through social collaborations)



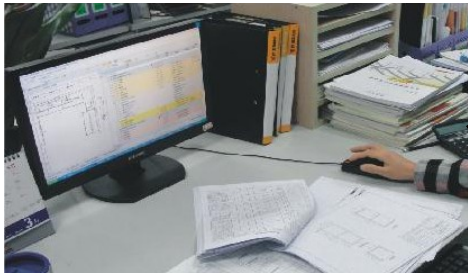
Online Shopping System



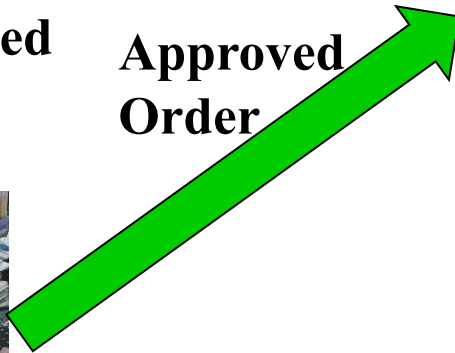
browse products
shopping cart management
confirm order
(by customer)



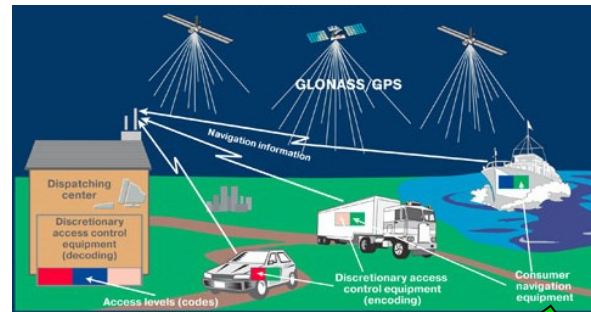
Submitted Order



order approval
(by online shop staff)



Approved Order



dispatching system

Dispatching and Navigation Instructions



Find place and deliver produces
(by delivery man)



Human Inside

As an Expert

- business expert
- design expert
- IT infrastructure expert



As a Component

As a User

As an Agent



Human Factors in Self-Adaptive Systems

- **Human as Expert**
 - **Capture expertise at different layers as knowledge base**
 - **Form multi-layered control loops at runtime**
- **Human as User**
 - **Monitor and learn the changing and personalized quality requirements**
 - **Adapt according to the learned user quality requirements**
- **Human as Component**
 - **As a part of human architecture interacting with software architecture**
 - **Can be adapted with software architecture for system-level optimization**
- **Human as Agent**
 - **Have its own goals and try to best satisfy the goals**
 - **Rely on its own capabilities and also social collaborations with others**

Human Factors in Self-Adaptive Systems

As an Expert

- business expert
- design expert
- IT infrastructure expert

As a Component



As a User

As an Agent

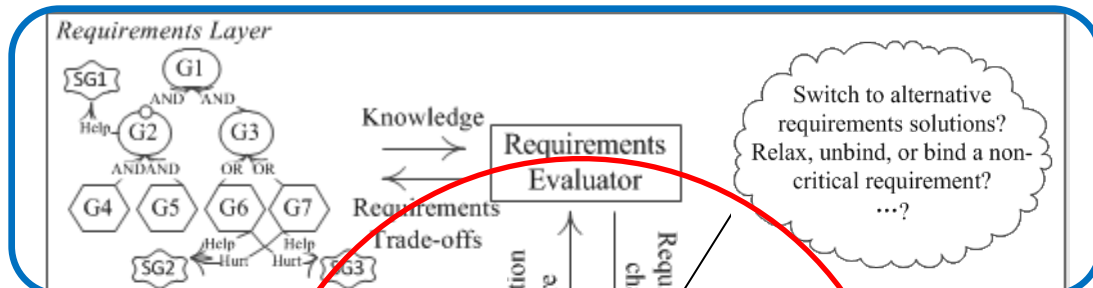


Human as Expert

- **Business Expert**
 - **Know how to optimize business processes and strategies in different situations**
- **Design Expert**
 - **Know different design options for the same issues (e.g., architecture styles and patterns, interaction protocols between components) and quality tradeoffs involved**
- **IT Infrastructure Expert**
 - **Know how to configure and optimize IT infrastructure such as network, middleware and virtual machine management**

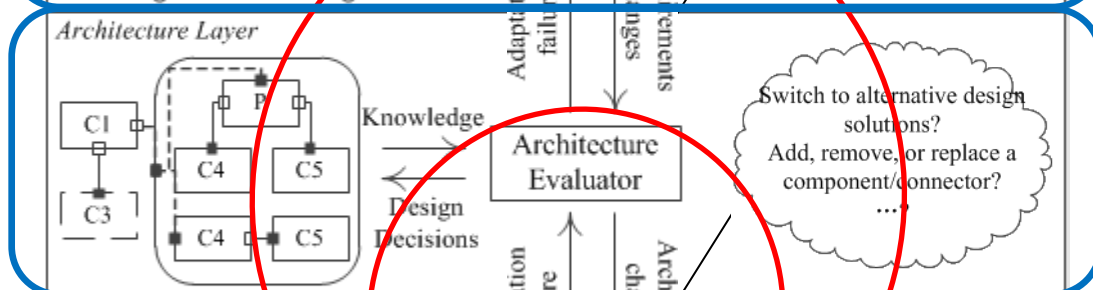
Multi-Layered Control Loops

Knowledge of Business Experts



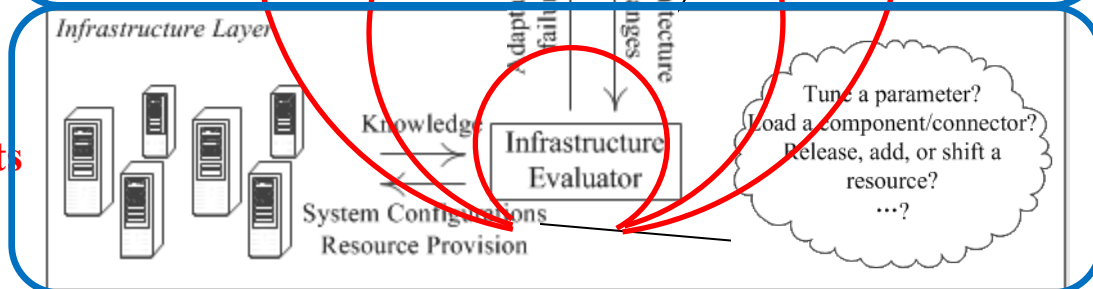
Operations on runtime requirements model

Knowledge of Design Experts



Operations on runtime architecture model

Knowledge of IT Infrastructure Experts



Management API provided by IT infrastructure



Interactions between Different Layers

- **Bottom-up: if problems cannot be handled by lower-layer adaptation, adaptation requests are thrown to higher layers**
 - **For example, when the performance issue can not be improved by infrastructure-layer adaptation, architectural reconfiguration or business process optimization may be launched**
- **Top-down: lower-layer adaptation policies can be adjusted by higher-level adaptation mechanisms**
 - **For example, the resource provisioning policies at the infrastructure layer may be adjusted by business layer**
 - ✓ **Allowed cost for renting virtual machines per hour: increased to \$100 when serving high-profit business; decreased \$50 at other time**

Human Factors in Self-Adaptive Systems

As an Expert

- business expert
- design expert
- IT infrastructure expert

As a Component



As a User

As an Agent



Personalized and Changing Quality Requirements

- Different users may have different experiences of the same quality requirement
 - **A response time of 10 seconds is unacceptable for an impatient young man doing online shopping but is good enough for an old man**
- A user may have different experiences of the same quality requirement at different time
 - **Doing a hurried shopping for a journey**
 - **Under an absent-minded and lazy condition**



Monitor and Learn User Quality Requirements

- Implicitly monitor user behaviors and feedback, for example
 - use eye tracking to detect the user's attention
 - use touch sensor to detect the frequency of user clicking
- Learn user quality requirements, e.g., a user's sensitiveness and endurance to a quality requirement



Adapt to Learned User Quality Requirements

- Analyze the minimum satisfaction levels that a user can accept
- Adapt to learned user quality requirements
 - **Optimized tradeoff decisions of multiple user quality requirements, e.g., lowering response time to save mobile phone energy**
 - **Save resources to better satisfy the quality requirements of other users, e.g., giving priority to the requests of the users who currently are more sensitive to response time**

Human Factors in Self-Adaptive Systems

As an Expert

- business expert
- design expert
- IT infrastructure expert

As a Component



As a User

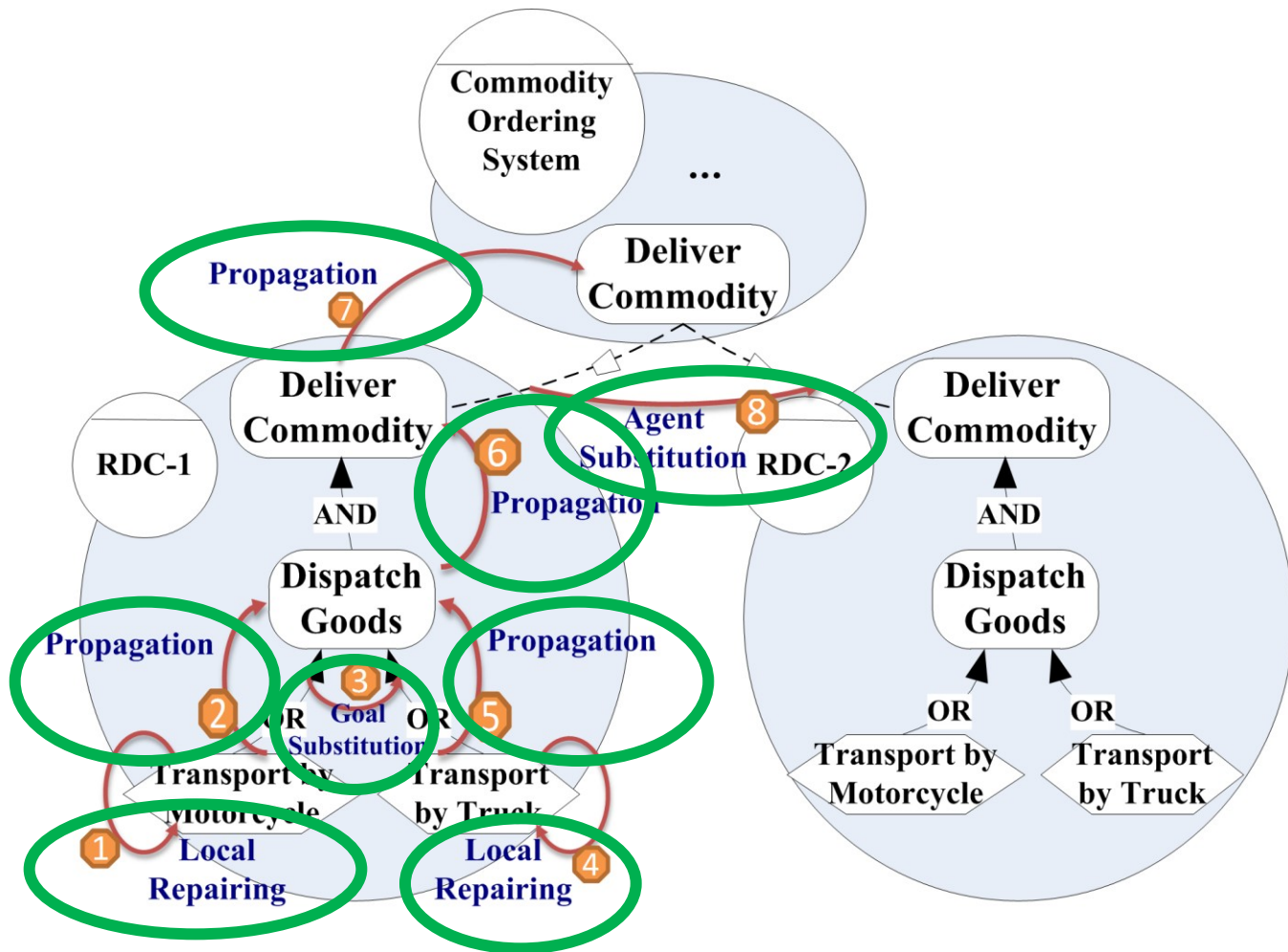
As an Agent



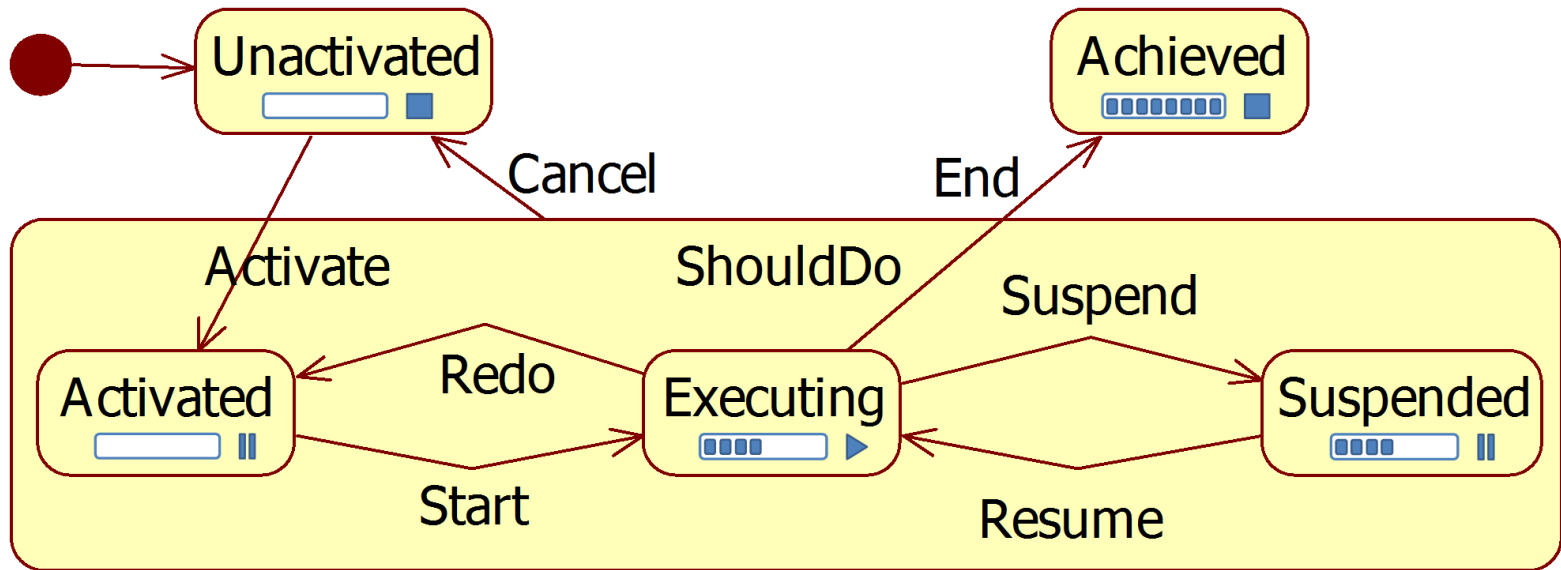
Agent-Oriented Adaptation in Socio-Technical Systems

- Decentralized requirements monitoring and adaptation
- Support social interactions among agents
 - **Capture and reason about the goals, capabilities, and social collaboration of each agent**
 - **Adaptable interactions**
 - ✓ agent substitution as an adaptation strategy
 - ✓ adaptable contracts and commitments
 - ✓ commitments can be delegated to other agents (e.g., friends, colleagues)

Hierarchical Repairing Process

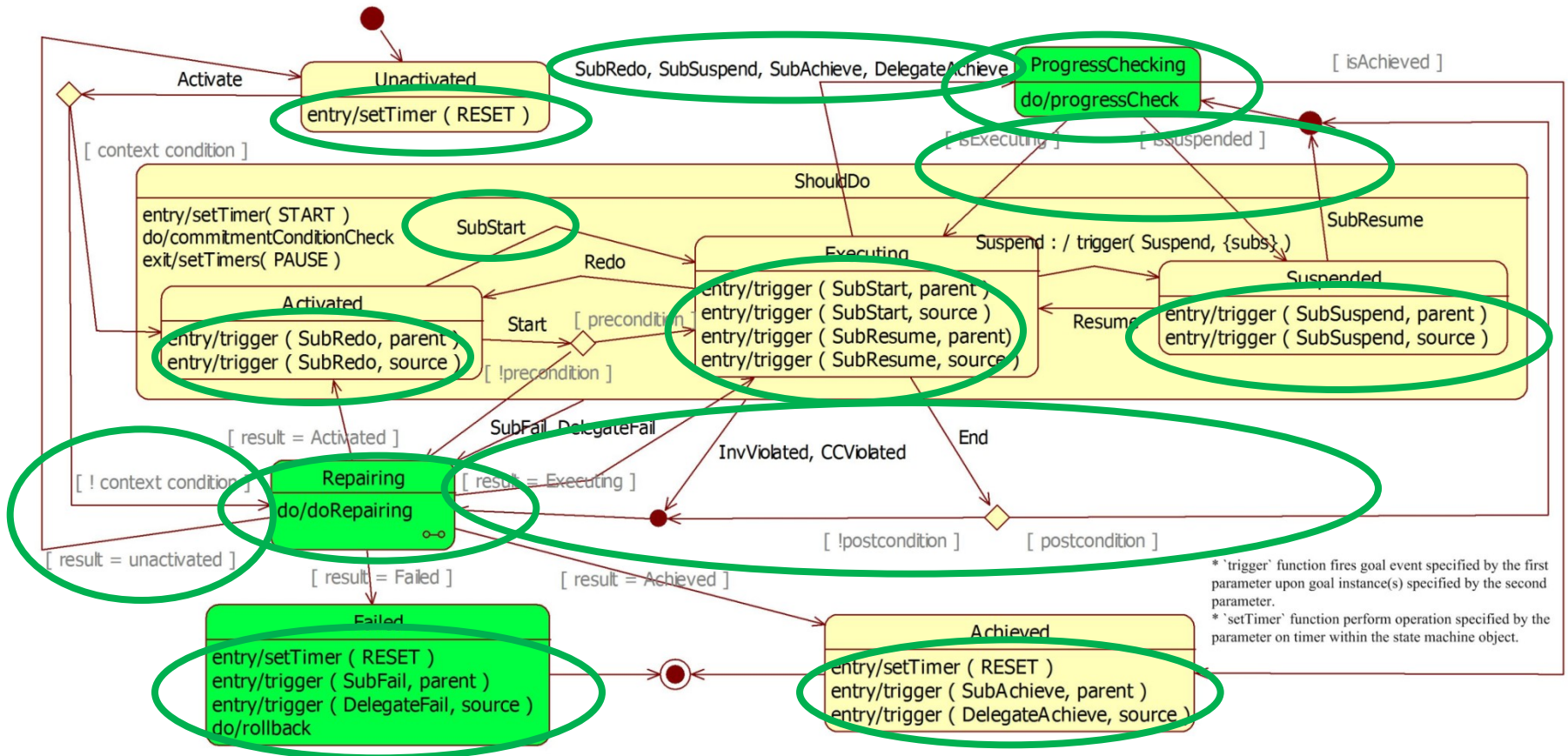


Goal State Machine



- Isomorphic state machines for all goals
- Each goal instance has its own goal state machine instance at runtime
- Goal state machine instances interact with each other via goal refinement and delegation relations

Extended Goal State Machine



- extend with states, actions and transitions for monitoring and repairing
- support inter-agent interactions by state machine interactions with messages



Commitment

- Treat socio-technical systems (STSs) as protocols consisting of a collection of commitments
- A commitment C is a tuple $(Debtor, Creditor, Antecedent, Consequent)$
 - ***Debtor* agent is committed to fulfill the *Consequent* condition for the *Creditor* agent if the *Antecedent* condition holds**
- The requirements for an STS are modelled as goals
- A specification for fulfilling the requirements (a.k.s. *protocol*) is modelled as a collection of commitments



Self-Adaptation with Evolving Commitments

- Initially established commitments may be renegotiated or cancelled during adaptation, given that compensation, delay, or cancellation are acceptable
- Possible adaptation alternatives should be considered and evaluated together with possible commitment changes or cancellation by each participating agent
- At all times, each agent selects the best adaptation strategy based on a cost-benefit analysis and makes decisions accordingly

Human Factors in Self-Adaptive Systems

As an Expert

- business expert
- design expert
- IT infrastructure expert

As a Component



As a User

As an Agent



Self-Adaptation Scenarios

- Scenario 1: navigation requests from delivery drivers grow fast, then adapt by switching to a more efficient navigation algorithm with lower accuracy
 - **The system needs to anticipate the effects to truck drivers: it is acceptable to tune the accuracy from 5 meters to 50 meters; but not acceptable if decreased to 500 meters.**
- Scenario 2: delivery delays and customer complains grow fast
 - **IT systems run well and order approval staff becomes bottlenecks**
 - **The system needs to know the bottleneck and adapt the human part, for example, by adapting the collaboration structure, calling more staff to join the work online**



Human as Component

- **Human Architecture:** describe the system's users in terms of human components and collaboration connectors along with their means of communication and coordination

[Christoph Dorn, Richard N. Taylor, ICSE 2013]
- **Software architecture concepts applied for humans**
 - **Interaction with connectors (remote or physical ways)**
 - **Architecture styles and patterns: publish-subscribe, P2P, ...**
 - **Quality attributes of human component: response time, throughput...**
- **Something different from software architecture**
 - **Human himself as component can be adapted to different roles within his capability**
 - **Human as component is more uncertain than software component**



Adaptive Human Architecture

- Model and reason about the organizational aspects
 - **collaborative relationships among individuals**
 - **Roles of individuals (responsibility, authority)**
- Reason about side effects of software adaptation to human agents
- Be aware of human users becoming bottlenecks
- To support adaptation decisions involving not only software adaptation but also human adaptation
 - **software adaptation (e.g., behaviors, computing resources)**
 - **human adaptation (e.g., roles, human resources)**



Conclusion

- Consider human factors from different perspectives for self-adaptive systems (SASs)
 - **Human as: expert, user, agent, component, ...**
- More about human in SAS: collaborative adaptation in social network
 - **Location based: find collaborators who are in the right positions**
 - **Human as sensor: collaborative sensing**
 - **Outsourcing goals: decompose and outsource**



Thanks!