

# Social Computing: Principles, Platforms, and Applications

Amit K. Chopra  
University of Trento

Workshop on Requirements Engineering for Social Computing, 2011

Aug 29, Trento

# Social Applications

## Interaction Among Autonomous Agents

- ▶ Business processes
  - ▶ Banking
  - ▶ Car insurance
  - ▶ Healthcare
- ▶ Social networks
- ▶ Argumentation
- ▶ Software engineering itself

# Social Dependence

## Commonality Across Social Applications

- ▶ Doctors depend on some civic body for salary
- ▶ One bank depends on another to settle transactions
- ▶ A friend depends on another not to share photos outside the circle
- ▶ General public depends on claims of scientific bodies that glaciers are melting ever faster
- ▶ Community depends on members' acceptance of what *counts as* what

# Social Computation

## Evolution of Social Dependence

The computers are autonomous agents

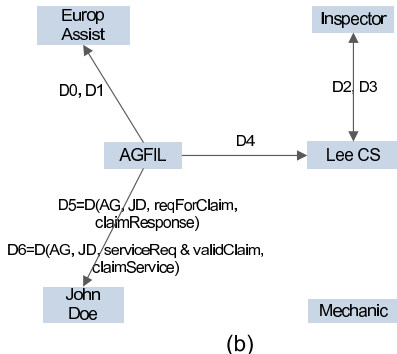
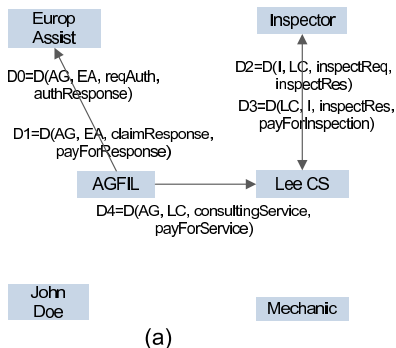
- ▶ Humans, organizations, or their software surrogates
- ▶ Each independently motivated (and designed)

As agents interact, social state evolves

- ▶ Each agent computes social state from its own local observations
  - ▶ No global state as such

# Social State Evolution

## Car Insurance Business Process



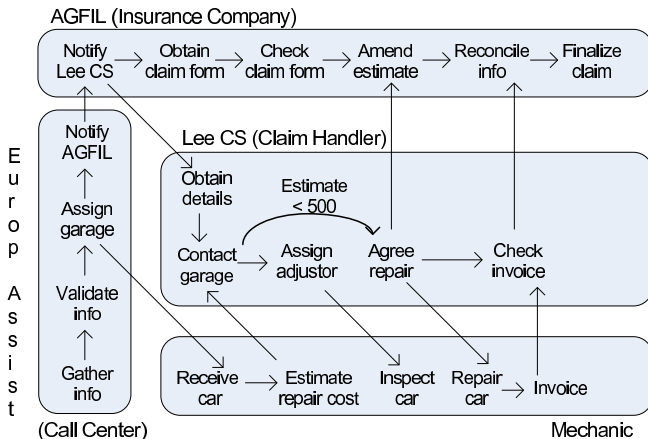
# Hypothesis

If we understood the nature of social dependence, many potentially diverse classes of applications could be built from the same high-level abstractions

- ▶ Simplifies their software engineering
- ▶ Common semantic basis

# Current Software Engineering

## Car Insurance Workflow



# Current Software Engineering

## Not Suited to Building Social Applications

Low-level abstractions based on control flow resulting in

- ▶ Overspecified systems
- ▶ Less reusability
- ▶ Less manageable designs and code
- ▶ Less interoperability across applications
- ▶ Social aspects handled offline

Commonality, if any, is at a lower-level



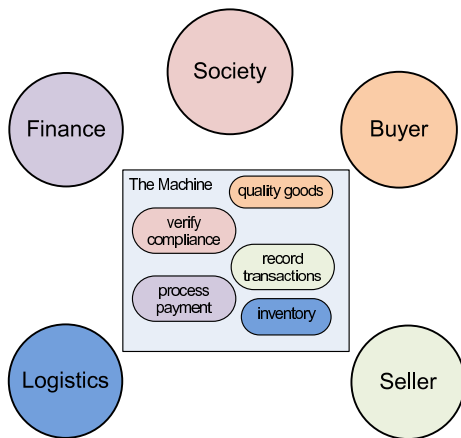
# Current Software Engineering

## Platforms for Social Applications

- ▶ Web: database abstractions
- ▶ WS-\*: layer on top the Web, but again lacks social abstractions

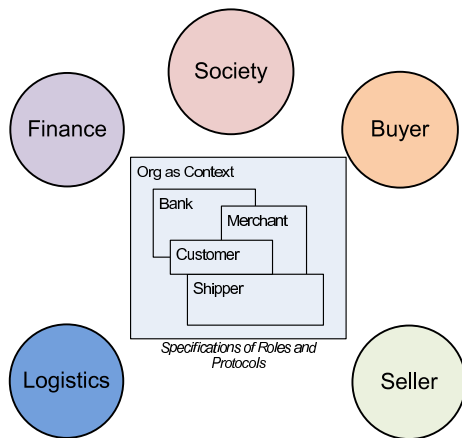
# Current Software Engineering

## Machine Orientation



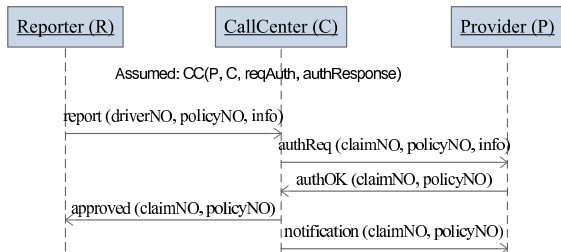
# Way Forward For Social Applications

## Interaction Orientation

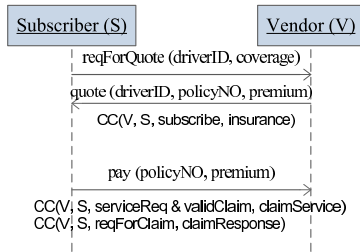


# Car Insurance Protocols

## Claim reception and verification (Rec)



## Insuring policy buyers (Ins)



# Challenge One: Social Abstractions

## Promising candidates

- ▶ Commitment: customer is socially committed to merchant for payment in return for goods
- ▶ Trust: doctors trust civic bodies to pay their salaries

Patterns over the elementary abstractions

# Challenge Two: Social Application Specification Language

Intuition: essentially in terms of interactions protocols

- ▶ Agents themselves are arbitrary

# Challenge Three: Software Engineering Principles

Do we need to revisit and reinforce them?

E.g.: Modularity

- ▶ Agents (roles) are the fundamental units of modularity in system decomposition
  - ▶ An agent's autonomy derives from that of its principal
  - ▶ Fail modularity: business workflows such as BPEL

# Challenge Four: Methodology

Do we need new ideas here?

Protocols and agents would be independently designed

- ▶ How we do design protocols and agents from stakeholder requirements?
- ▶ What kinds of reasoning and tools would best support their design?



# Challenge Five: Social Platform

## Distributed Enactment

Provides infrastructure services

- ▶ Would support the primitive interaction protocols
- ▶ Social API for programming agents
- ▶ Discovery services
- ▶ Social-level interoperability

# Social Computer

## Is Not Social Computing

### Is a machine

- ▶ Solves social problems by considering inputs from social entities (including other social computers) in light of social conventions
  - ▶ Google's PageRank: lightweight social computer if one considers a link as a vote

# Conclusions

Current SE approaches emphasize control instead of interaction

- ▶ Mismatch with the nature of social applications

Social computing emphasizes interaction and the computation of social relationships

- ▶ An approach for specifying, implementing, and enacting social applications
- ▶ Provides a common semantic basis and platform for many diverse kinds of applications

# Acknowledgments

- ▶ Munindar Singh, Fabiano Dalpiaz, John Mylopoulos, Paolo Giorgini, Matteo Baldoni, Nicolas Maudet, Pinar Yolum, Michael Huhns, and Michael Jackson
- ▶ Marie Curie Trentino Cofund