The Social Compute Unit: Adaptability and Rewarding Mechanisms

Ognjen Šćekić, Schahram Dustdar

Distributed Systems Group
Institute of Information Systems

www.infosys.tuwien.ac.at
Outline

- Introduction & Motivation
- The Social Compute Unit (SCU)
- Rewarding, Incentives and Adaptability
- Evaluating Results
Introduction
Introduction – What is out there?

- **Crowdsourcing** - human collaboration model. Problem presented to an unstructured, large group of people. The problem is typically simple, and often naively parallel.

- **Social Computing** - more general approach. Ad-hoc assembled team of experts to solve a particular problem. Team members are picked based on their expertise and existing relations from a social/professional network they already belong to.

- **Mixed Systems** - transparent compositions of traditional software services and human-based computation, interfaced via service model (e.g. via Human-Provided Services).

- **The Social Computing Unit (SCU)** - a novel construct that goes one step further from just assembling teams from appropriate experts. Looks into the entire lifecycle of human expert teams in the context of Mixed Systems.
Introduction – Motivating Problem

- New social computing solutions to complement traditional business processes.

- Better support for ad-hoc assembled, structured interactions and work styles suited for today’s global business reality.

- SCU - essential novelties of team programmability and elasticity.

- Program and “compile” virtual teams of resources with appropriate skills in the problem domain.
- Integration with BPM/workflow technologies to achieve efficient mixed systems.
The Social Compute Unit
Social Compute Unit (SCU) Lifecycle

Principal SCU lifecycle phases:

- **Team creation**
  - Assembly of requested human expert teams.
  - Contributors "compiled" to match the required skills, problem domain and envisioned team structure/hierarchy.

- **Task execution**
  - Iterative and parallel work patterns.
  - Adjust reward scores for each contributor and adapt the team.

- **Team dissolution**
  - Honoring of contractual obligations between SCU and contributors.
  - Paying out of rewards or penalties depending on accumulated scores.
SCU – Execution Phase

- Iterative and parallel execution patterns.
- Tunable iteration cycles (duration, number of participants, QoS, etc.)
- Good trade-off between **stability** necessary to perform complex assignments and **adaptability** needed to counteract the inherent instability of human participation.
- Simplified monitoring.
- At the end of iteration: adjust scores and re-compile (adapt) the team.
Rewarding, Incentives and Adaptability
**Definitions**

- **Incentives**
  Stimulate (motivate) or discourage certain worker activities before the actual execution of those activities.

- **Rewards**
  Any kind of recompense for worthy services rendered or retribution for wrongdoing exerted upon workers after the completion of activity.

- **Incentive/Rewarding Scheme**
  A set of rules and mechanisms for assigning rewards and incentives.
SCU – Metrics for Human QoS

- Difficulty of work
- Expertise
- Experience
- Willingness
- Required working time
- Actual working time
- Unit price
- Quality of work
- Effort level
- Average response time
- Effectiveness
- Total iteration execution time
- Total iteration price
- Trust
Language for Rewarding & Adaptability

- Exploring and defining rewarding strategies is a separate question from creating a general-purpose mechanism for performing rewarding.
- Need a rule-based language that takes as an input a topology (a team structure), and applies the rules on the graph nodes.
- Applying the rules should follow some general guidelines, to maintain some invariants, that can be either general or team-specific.

- Applying the rules proceeds in following phases:
  1. At the end of each iteration update (calculate) the metrics
  2. Identify the nodes affected by a rule
  3. Adjust/update the score based on the rule
  4. Repeat 2-3 until all rules exhausted
  5. Decide on a possible team reconfiguration
  6. Apply incentive policies
Language for Rewarding & Adaptability

- Types of rules:
  1. Rewarding rules
  2. Incentive rules
  3. Adaptation (reconfiguration) rules

- Each rule has a part that identifies the intended targets.
  - Based on some graph properties and/or attributes that the nodes possess
  - e.g. All leaf nodes that have attribute: `role="testEngineer"` and `score > k`

- Each rule should have a part that specifies what action should be taken against the targeted nodes.
  - Actions are user-defined separately
  - e.g. Apply actions: `(RewardThoseThatFinishedInTime AND RewardBestWorkerInThisCycleWithPlus10Percent AND MotivateNewcomers) OR ApplyRandomAward`
Language for Rewarding & Adaptability

- Actions are defined separately.
- They exert an action on the attributes (including the score) of each node in the set of matched input nodes.
- Definition of an action contains the code that will directly alter the attributes.
- Actions are used in both incentive and rewarding rules.
- Rules are executed in order, meaning that the effect of later rules can affect the execution of previous rules.
Language for Rewarding & Adaptability

- Each node remembers the rules that were applied onto it, and their order.

- Exclusive or cumulative application of the rules.

- Reconfiguration rules are evaluated after rewarding rules.

- Reconfiguration rules marking the nodes with different flags.
Flags - used to mark the nodes/edges to be discharged or added, and contain the instructions on how to reconfigure the team.

- Honorable or dishonorable discharge.

- Flags are not user-defined. Therefore, team reconfiguration will be handled by the system at this stage.
Evaluating Results
Evaluating Results

- Evaluation of adaptability mechanisms
  - Simulating humans as software agents, providing web services.
  - Simulations to be performed on a framework for large-scale service testbed generation - Genesis 2.

- Evaluation of rewarding and incentive mechanisms
  - Problematic. Still considering the best approaches here.
  - Current research focused mostly on crowdsourcing.
  - Social simulation techniques and games involving our students.
  - The evaluation metrics: reward money, productivity, drop-out rate, etc.
Thank you for your attention!

Ognjen Šćekić
Distributed Systems Group
Institute of Information Systems
Vienna University of Technology
Austria

http://www.infosys.tuwien.ac.at/staff/phdschool/oscekic/