One-to-Many Multi-agent Negotiation and Coordination Mechanisms to Manage User Satisfaction

Amro Najjar, Yazan Mualla, Kamal Singh, Gauthier Picard

14 July 2018
ACAN@AAMAS 2018, Stockholm, Sweden
Satisfaction Management (SM)

- The provider seeking a balance between:
  1. Minimizing its costs and use its resources efficiently
  2. Providing a (fine-grained) satisfactory service to its users
Satisfaction Management

- The provider seeking a balance between:
  1. Minimizing its costs and use its resources efficiently
  2. Providing a (fine-grained) satisfactory service to its users
Satisfaction Management

Satisfaction Management (SM)

- The provider seeking a balance between:
  1. Minimizing its costs and use its resources efficiently
  2. Providing a (fine-grained) satisfactory service to its users

Problem!

In most of existing works, SM decision is taken unilaterally by the provider, the end-user preferences and her expectations are overlooked, inter-user differences are ignored
A one-to-many negotiation architecture whose goals are:

**Objectives**

- Integrate the user’s subjective evaluation of the service quality [Hoßfeld et al., 2016, Najjar, 2015]
- Equip the provider with a fine-grained control over end-user satisfaction while respecting its budget constraints
Proposition

A one-to-many negotiation architecture whose goals are:

**Objectives**

- Integrate the user’s subjective evaluation of the service quality [Hoßfeld et al., 2016, Najjar, 2015]
- Equip the provider with a fine-grained control over end-user satisfaction while respecting its budget constraints

In order to understand and model users’ satisfaction and their subjective acceptability thresholds, we resorted to the literature of:

**Related Domains**

1. Customer expectation management [Zeithaml et al., 1993] and Psychophysics [Reichl et al., 2010]
2. Quality of Experience (QoE) [Möller and Raake, 2014]
Plan

1. Context
2. Quality of Experience
3. One-to-Many Negotiation
4. Adaptive-QoE-Aware elasticity Management (AQUAMan*)
5. Evaluation
Quality of Experience

- QoE: is the service quality estimated **subjectively** by the end-user (a subjective and user-centric measure)
Quality of Experience

- QoE: is the service quality estimated **subjectively** by the end-user (a subjective and user-centric measure)

**Challenge**

Most of existing works are *unilateral* and they rely on MOS (Mean Opinion Score). Yet, MOS hides information about user diversity and ignores users evolution [Hoßfeld et al., 2011]
Quality of Experience

- QoE: is the service quality estimated **subjectively** by the end-user (a subjective and user-centric measure)

**Challenge**

Most of existing works are *unilateral* and they rely on MOS (Mean Opinion Score). Yet, MOS hides information about user diversity and ignores users evolution [Hoßfeld et al., 2011]

**A Possible Solution**

- Recent empirical studies on QoE recommend providers to rely on **percentiles** to evaluate the user satisfaction [Hoßfeld et al., 2016]
- Agents can integrate user’s **personal and evolving** preferences and involve her in the decision-making process [Najjar et al., 2017, Najjar et al., 2016a]
Plan

1. Context
2. Quality of Experience
3. One-to-Many Negotiation
4. Adaptive-QoE-Aware elasticity Management (AQUAMan*)
5. Evaluation
One-to-Many Negotiation

- Allows a seller to negotiate simultaneously with several buyers [Nguyen and Jennings, 2004]
- Involves negotiation and coordination strategies [Rahwan et al., 2002]
One-to-Many Negotiation

- Allows a seller to negotiate simultaneously with several buyers
  [Nguyen and Jennings, 2004]
- Involves negotiation and coordination strategies
  [Rahwan et al., 2002]

Challenges
- Most of existing works address atomic simultaneous negotiation sessions
- One agreement is required, all other sessions are aborted
  - Useful works in service composition
    [Mansour and Kowalczyk, 2014, Richter et al., 2012]
1. Context
2. Quality of Experience
3. One-to-Many Negotiation
4. Adaptive-QoE-Aware elasticity Management (AQUAMan*)
5. Evaluation
Integrate the user into the loop [Najjar et al., 2016a]

Allows for QoE-aware and personal elasticity management [Najjar et al., 2016b]

End-users have personalized utility functions and negotiation strategies

Negotiation sessions are autonomous but they must respect their budget constraints RC

Users can enter the system and leave at will (with variable influx)
AQUAMan*: User Satisfaction Management

During the negotiation process, user can accept/reject service depending on their expectations and subjective evaluation of its quality

AQUAMan’s Adaptive Mechanism

► **Goal**: guarantee a predefined satisfaction goals while not exceeding $RC$, the average cost invested per user

► **Solution**: to do so, the provider must

1. Construct a model of negotiation behavior of each user
   
   [Najjar et al., 2017, Baarslag et al., 2015]

2. Adjust its negotiation strategy in order to restore user satisfaction to its predefined goals while $RC$
Formalization

**Satisfaction Goals**

- Enforce Goal\([k]\) how much percent of users receive a service of the satisfaction category \(k\) (e.g. acceptable, good, excellent, etc.)

**Satisfaction Management as an Optimization Problem**

Minimize

\[
\sum_{k=1}^{K} \sum_{i=1}^{N^t} C_i^k \cdot X_i^k
\]  

Subject to

\[
\sum_{k=1}^{K} X_i^k \leq 1, \quad \forall i
\]  

\[
\sum_{m=1}^{k} \left( \frac{\sum_{i=1}^{N^t} X_i^m + \text{Already}[m]}{\# \text{TerminatedSessions}} \right) \geq \text{Goal}[k], \quad \forall k
\]
Heuristic Algorithm

Construct a list of users approaching their deadline

**Sorting the list**

1. $\textit{Cost}_{First}$: users with cheaper acceptability requirements are put on top of the list
2. $\textit{Utility}_{First}$: users with cheaper satisfaction on top
3. $\textit{Deadline}_{First}$: the shorter the user’s remaining time is, the upper in the list it is

**Selecting Users from The list**

1. $\textit{DISTANCE\_TO\_GOAL}$: the category furtherest from achieving its satisfaction goals is prioritized
2. $\textit{CATEGORY\_COST}$: the category the less costly is prioritized

Based on the user model, propose a tailored offer
1. Context
2. Quality of Experience
3. One-to-Many Negotiation
4. Adaptive-QoE-Aware elasticity Management (AQUAMan*)
5. Evaluation
**Optimal Solution vs Heuristic Algorithm**

![Graph showing comparison between Optimal Solver and Heuristic Algorithm](image)

**Table: Heuristic strategies vs. optimal cost**

<table>
<thead>
<tr>
<th>Category/Cost</th>
<th>Utility First</th>
<th>Cost First</th>
<th>Deadline First</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>DISTANCE_TO_GOAL</strong></td>
<td>+10.2%</td>
<td>+16.5%</td>
<td>+10.9%</td>
</tr>
<tr>
<td><strong>CATEGORY_COST</strong></td>
<td>+12.9%</td>
<td>+12.9%</td>
<td>+13.1%</td>
</tr>
</tbody>
</table>

**Results**

- The optimizer is intractable with high number of users
- Cost-wise, the best heuristic is 10% more expensive than the optimal solution
**User Satisfaction & Cost**

**Figure:** The user satisfaction achieved by the heuristic strategies (red curves) and their cost (blue dashed curve) compared with the optimal solution cost (blue solid curve).
Impact of User Feedback

Figure: The impact of $f\%$ (x-axis) on the ratio of users assigned to each satisfaction category (left y-axis) and on the cost spent per served user (right y-axis).
Impact of BBC Users

Figure: The Impact of BBC Users.
Conclusions

- AQUAMan* equips the user with fine-grained control over user satisfaction
- Satisfies the provider cost constraints
- Executable in tractable time

Future Works

- Study the relations among different satisfaction goals
- Introduce measures of QoE fairness
- Develop BBC Handling Strategies
Conclusions & Future Works

**Conclusions**

- AQUAMan* equips the user with fine-grained control over user satisfaction
- Satisfies the provider cost constraints
- Executable in tractable time

**Future Works**

- Study the relations among different satisfaction goals
- Introduce measures of *QoE fairness*
- Develop BBC Handling Strategies

In QoMEX, pages 131–136.


Springer International Publishing.


We set \( \text{Goal} = 100\% \) to push AQUAMan to accept as much user as possible in order to identify its limit.
The cost spent per user of AQUAMan compared with other non-adaptive strategies
The negotiation object is offer $o^t_i = \langle v_1, v_2, \ldots, v_J \rangle$

- It assigns a value $v_k$ ($k \in [1, J]$) to each of the service attribute.

Example: in the transcoding service an offer is:

$$o^t_{da_i} = \langle HD, 10 \text{ min}, 24p \rangle$$
Concession Strategy

- $s_{ai}$ make concessions by reducing their aspiration rate
- $\Delta AR_t^i = AR_t^i \cdot \left(\frac{t}{T_{sa_i}}\right)^{\lambda_i}$
- Then $AR_{t+1}^i = AR_t^i - \Delta AR_t^i$
- When $T_{sa_i}$ is reached, $s_{ai}$ quits the negotiation

$\lambda$ controls the convexity of the concession curve
- $\lambda_i < 1$: Conciliatory
- $\lambda_i \approx 1$: linear
- $\lambda_i > 1$: Boulware
saᵢ rely on a utility function $M$ derived from $suᵢ$ preferences

- $M$ is a weighted sum of $μ_{saᵢ}$, linear attribute utility functions

$$μ_{saᵢ,at_j}(o^t_{daᵢ}[at_j]) = \frac{rv_{saᵢ,at_j} - o^t_{daᵢ}[at_j]}{rv_{saᵢ,at_j} - pv_{saᵢ,at_j}}$$ (4)

- $rv_{saᵢ,at_j}$ and $pv_{saᵢ,at_j}$ are the preferred and reservation values of the attribute $at_j$

- $saᵢ$'s subsequent decision is based on the acceptance condition
QoE is a function of influence factors. Influence factors are the independent variables and QoE is dependent variable:

\[ QoE_s = \Phi(IF_1, IF_2, \ldots, IF_n) \]
QoE Influence Factors

QoE is a function of influence factors. Influence factors are the independent variables and QoE is the dependent variable:

\[ \text{QoE}_s = \Phi(\text{IF}_1, \text{IF}_2, \ldots, \text{IF}_n) \]

1. **SIF**: technical aspects of the system on which the user consumes the factors.
2. **HIF**: user background, expectations, his personality and previous experience.
3. **CIF**: the context when the user consumes the service.
QoE-aware Agent Decision Model

- The user agent seeks to maximize the **subjective satisfaction** of the user.

- Therefore, it derives its decision model and utility function from the user preferences.
  - Thus, for each attribute:
    \[ \mu_{s_a, at_j} = \alpha_{s_a, at_j} \cdot f(v_{at_j}) + \beta_{s_a, at_j} \]  
      \[ (5) \]

- The form of \( f() \) depends on the attribute type.

- If the service involves multiple attributes:
  \[ M_{s_a}(o) = \sum_{j=1}^{J} w^t_{s_a, j} \cdot \mu_{s_a, at_j}(o^t_{da_i}[at_j]) \]  
    \[ (6) \]

- Where \( o \) is an offer received from the provider.
The Logarithmic Hypothesis

- Derived from the Weber Fechner Law (WFL) of psychophysics
- Logarithmic relationship between QoS parameter and QoE:

\[ QoE = -\alpha \cdot \log(QoS) + \beta \quad (7) \]

- QoS should be perceivable by the user (e.g. waiting time)
- Validated and applied to various applications (file download, Email, etc.)
$M$ becomes a weighted sum of logarithmic attribute utility functions:

$$M_{sa_i}(o_{da_i}^t) = \sum_{j=1}^{j=J} w_{sa_i,j} \cdot [-\alpha_{sa_i,at_j} \cdot \ln(o_{da_i}^t[at_j]) + \beta_{sa_i,at_j}]$$ (8)

**Coefficients**

- $\alpha_{sa_i,at_j}$, $\beta_{sa_i,at_j}$ are the personal coefficients of $sa_i$ for $at_j$
- Derived from $sa_i$ preferences:

$$\alpha_{sa_i} = \frac{1}{\ln(rv_{i_{res}}) - \ln(pv_{i_{res}})} \quad \beta_{sa_i} = \frac{\ln(rv_{i_{res}})}{\ln(rv_{i_{jct}}) - \ln(pv_{i_{res}})}$$ (9)
### Provider (ASP) Side

- A SaaS provider whose service is compute-intensive
- Rents resources from a cloud provider (CP) (e.g. Amazon EC2) to fulfill the requests
Use-case scenario: Cloud-hosted Service

Provider (ASP) Side
- A SaaS provider whose service is compute-intensive
- Rents resources from a cloud provider (CP) (e.g. Amazon EC2) to fulfill the requests

User Side
- We suppose that all attributes of the service are directly perceivable by the user (e.g. execution time)
- Therefore, we consider that the logarithmic hypothesis applies to these attributes

\[
\mu_{sa_i,at_j}(v_{at_j}) = -\alpha_{sa_i,at_j} \cdot \ln(v_{at_j}) + \beta_{sa_i,at_j}
\]  (10)
States the rules of interaction between ca and da
proposed to satisfy the specific requirements of elasticity management negotiation
Illocutions sent from da to ca are different from those sent from ca to da

<table>
<thead>
<tr>
<th>Illocutions (da → ca)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Inform(event)</td>
<td>da notifies the coordinator ca about an important event</td>
</tr>
<tr>
<td>RequestIntervention</td>
<td>da asks ca to intervene in its session i because the decision is beyond da capacities.</td>
</tr>
</tbody>
</table>
The illocutions sent from the ca to da\textsubscript{i} are imperative

da\textsubscript{i} cannot reject it

<table>
<thead>
<tr>
<th>Illocutions (ca → da\textsubscript{i})</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Spawn (Ω\textsubscript{da\textsubscript{i}})</td>
<td>ca spawns a delegate da\textsubscript{i} to negotiate with sa\textsubscript{i}</td>
</tr>
<tr>
<td>Modify(Ω′\textsubscript{da\textsubscript{i}})</td>
<td>ca alters the negotiation strategy of da\textsubscript{i}.</td>
</tr>
<tr>
<td>Suspend</td>
<td>ca orders the delegate da\textsubscript{i} to suspend the negotiation process in the session (i).</td>
</tr>
<tr>
<td>Kill</td>
<td>ca terminates the delegate da\textsubscript{i}</td>
</tr>
</tbody>
</table>
Example

Bob $\mu(jct)$

Alice $\mu(jct)$

preferred JCT

reservation JCT

preferred JCT

reservation JCT
 Example

\[ \mu_{s_{i,j}}(jct) \] is a decreasing logarithmic function

- Bob, gold SU, has lower reservation and preferred values
- Alice, bronze SU, has higher reservation and preferred values
Example (2)

- Alice, bronze SU, has lower reservation and preferred values.
- Bob, gold SU, has higher reservation and preferred values.

The overall utility function, $\mu_s$, is a weighted sum of $\mu_s\text{res}$ and $\mu_s\text{jct}$. 

Graphs show the relationship between reservation and preferred resources for Bob and Alice.
Example (2)

▶ $\mu_{sai, res}$ is an increasing logarithmic function

▶ Bob, gold SU, has higher reservation and preferred values

▶ Alice, bronze SU, has lower reservation and preferred values

▶ $M_{sai}$, the overall utility function, is a weighted sum of $\mu_{sai, res}$ and $\mu_{sai, jct}$
Simple example of a performance model of a transcoding service

<table>
<thead>
<tr>
<th>$t/q$</th>
<th>SD</th>
<th>MD</th>
<th>HD</th>
</tr>
</thead>
<tbody>
<tr>
<td>10 min</td>
<td>4</td>
<td>5</td>
<td>8</td>
</tr>
<tr>
<td>20 min</td>
<td>2</td>
<td>4</td>
<td>6</td>
</tr>
<tr>
<td>30 min</td>
<td>1</td>
<td>2</td>
<td>3</td>
</tr>
</tbody>
</table>

- Tradeoffs
- Faster process requires more cloud resources $\implies$ more costly for the ASP