An Adaptive Mediation Framework for Mobile P2P Social Content Sharing

Chii Chang[1], Satish Narayana Srirama[2], and Sea Ling[1]

[1] Faculty of Information Technology, Monash University, Australia
{chii.chang;chris.ling}@monash.edu
[2] Institute of Computer Science, University of Tartu, Estonia
srirama@ut.ee

Abstract. Mobile Social Network in Proximity (MSNP) represents a new form of social network in which users are capable of interacting with their surroundings via their mobile devices in public mobile peer-to-peer (MP2P) environments. MSNP brings opportunity to people to meet new friends, share device content, and perform various social activities. However, as the fundamental topology of MSNP is based on public MP2P network, many challenges have arisen. Existing related works restrict the MP2P social network to operate in specific platforms and protocols. Enabling MSNP in a dynamic public MP2P requires a more flexible solution, which can adapt its behaviour to comply with environment. Hence, we propose a mobile device-hosted service-oriented workflow-based mediation framework for MSNP. The fundamental portion of the framework is based on the Enterprise Service Bus architecture which supports changes in runtime resources without the need to re-launch the application. In order to adapt to different situations, our workflow tasks adjust the execution behaviour at runtime. The workflow engine dynamically selects the best approach to complete the mobile user's request based on the cost and performance, calculated by combining fuzzy set and cost performance index. The developed prototype is discussed along with detailed performance.

1 Introduction

The evolved mobile technologies provide users convenient ways to participate in various virtual online social networks (OSN) such as Twitter [32], Facebook[13]. In the past few years, researchers [36,25,31,28,27,17] have tried to leverage OSN with short range mobile communication technologies (e.g., Bluetooth [6], Wi-Fi Direct [35]) to bring OSN activities into mobile peer-to-peer (MP2P) network. These new breeds of mobile social network (MSN) applications encourage users to socialise with people in proximity via their smart mobile devices, and potentially bring opportunities to make new friends. We use the term — Mobile Social Network in Proximity (MSNP) to illustrate such an environment in which participants are capable of performing various generic OSN activities with proximal users. A typical activity in MSNP is content (e.g., text, images, music, etc.) sharing and mashup [19].

C. Liu et al. (Eds.): ICSOC 2012, LNCS 7636, pp. 374–388, 2012.

Mashup is a content-driven composition technique used to compose content derived from various sources into a single customisable presentation. In MSNP, participants may generate various content from their smart mobile device and post/synchronise to different social websites (e.g., Twitter, Facebook, etc.) or cloud storages (e.g., Dropbox [12]). Let us call these participants *content providers*. A *content provider* may intend to share his/her new content to public proximal mobile users in order to bring more visitors/readers to his/her own web pages or potentially establish connection with new friends. The content provider's MSNP application may generate a metadata and advertise it to other MSNP participants' devices based on their preference. Meanwhile, some MSNP participants may also intend to perform a location-based content mashup from proximal content providers to retrieve their interested information. MSNP is useful for attendees to fast share information in the high population event such as Comiket [9] without establishing a centralised system in the venue.

Considering the resource limitations of mobile devices and the dynamic nature of MP2P environment, communication becomes a crucial challenge to both content provider and content consumer. In order to enhance the overall performance of MSNP communication, some tasks such as semantic service/content matchmaking process may be distributed to remote Cloud services (e.g., Google App Engine (GAE) [14], Amazon EC2 [2] etc.). However, distributing tasks to Cloud is not always an efficient solution, because utilising Cloud service consumes extra costs such as network latency, price of using the service etc. In some cases, remaining the communication within local wireless network is more efficient when both performance and cost are considered, especially when there are only a few MSNP peers involved. On the other hand, when there are many MSNP peers involved, it may be more efficient to distribute more tasks to more powerful Cloud services. Hence, there is a need to design a framework which is capable of dynamically change its approach at runtime to adapt to different situations, while the MSNP peer is performing MP2P social network activities.

In this paper, we propose AMSNP: an **A**daptive **M**ediation framework for service-oriented mobile **S**ocial **N**etwork in **P**roximity. The contributions consist of:

- A workflow-controlled Web service-oriented mediation framework for mobile devices to easily leverage heterogeneous service resources automatically.
- An adaptation scheme, which can automatically decide what services should be used to complete the workflow tasks. The decision making is based on a cost-performance index scheme.
- A prototype implementation, evaluated on a real mobile device.

The remainder of this paper is structured as follows: In Section 2, we summarise the foundation of MSNP, followed by our proposed framework, and the adaptiation strategy. Section 3 provides an example of how the MSNP activity can be modelled using workflow. Section 4 describes the prototype implementation and the evaluation results. In Section 5, we describe the difference between our work and related works. Section 6 provides the conclusion and future research direction.

2 System Design

2.1 Overview of MSNP

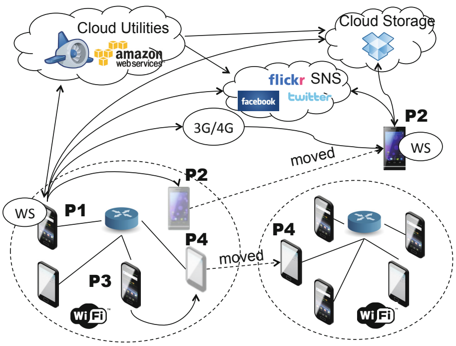


Fig. 1. MSNP architecture

In an MSNP environment, each mobile device is a mobile Web service consumer and also a provider [29]. When two peers join the same wireless network, they utilise standard communication technologies such as DPWS [22], or Zeroconf [15] to exchange their service description metadata (SDM). We expect each peer has its own backend Cloud storage to synchronise its IP address as a small text file in its Cloud storage (or alternatively utilising public DNS servers if available). The URL of the text file is described in a peer's SDM. Hence, when a peer (e.g., Fig.1, P2 and P4) moves out from the current network, the other peers (e.g., Fig.1, P1 and P3) in its previous network can still interact with it via mobile Internet.

Since P1 and P3 have previously exchanged their SDM with P2 and P4, they have cached the SDM of P2 and P4 in either their local memory or synchronised to their Cloud storages. When P1 and P3 receive requests from other peers in the same network that are performing service discovery, P1 and P3 can also provide P2 and P4's SDM to these peers. Instead of having the SDM directly send to the peers by P1 and P3, P1 and P3 can synchronise the cached SDM to their Cloud storages, and simply provide the URL link to the other peers. Similar concept is applied to content sharing and mashup, say for example, P1 intends to mashup the content provided by P2 and P3. When P1 invokes P2 and P3 for the content, P2 and P3 will simply reply the corresponding metadata documents, which contain the description about where the content can be retrieved from the Internet.

Taking into account that mobile devices usually have limited processing power, it is reasonable for a MSNP peer to delegate the processes to its backend Cloud utility service (CloudUtil). In Fig. 1 for example, P1 utilises its backend CloudUtil for semantic service discovery. Furthermore, CloudUtil can be used to directly access the content uploaded by other MSNP peers in Social Network Services (SNS) to discover useful content for P1's mashup (if the content has been described in Really Simple Syndication (RSS) feed format).

A content provider in MSNP can also actively push recommendation to other participants based on the participant's service preference. Due to privacy concerns, MSNP peers may prefer not to share their private information. However, when a list of available services (described semantically) is provided to the participant, the participant can simply reply which service type it is interested in.

This process can be done automatically by applying context-aware prefetching scheme, which has been described in our previous work [7].

2.2 AMSNP Framework

The framework design is based on the Enterprise Service Bus (ESB) architecture [26]. ESB is a software infrastructure that can easily connect resources by combining and assembling services to achieve a Service Oriented Architecture (SOA). Fig. 2 illustrates the architecture and main components of AMSNP. The architecture consists of four parts:

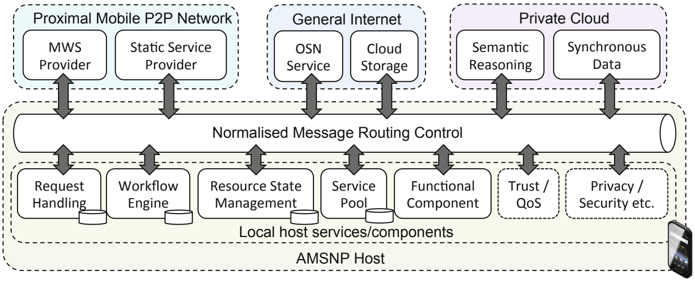


Fig. 2. Architecture of AMSNP Framework

- Proximal Mobile P2P Network — It represents the other MSNP peers within the same network. Depending on the developer's preference, an AMSNP host can support various network communication protocols such as XMPP [30], UPnP [33], Bonjour [4], etc.
- General Internet — Basically, the content generated by the MSNP peers are updated to their OSN services such as SNSs (e.g., Facebook, Twitter) or their Cloud storages. In our design, the Cloud storage services play an important role in MSNP. As mentioned previously, each MSNP peer synchronises its current IP address to its Cloud storage in order to resolve the dynamic IP issue of mobile P2P network.
- Private Cloud — MSNP peer can utilise a number of backend Cloud utility services for distributing tasks in order to reduce the resource usage of the device and also improve the overall performance. For example, the semantic service discovery process requires the MSNP peers to process a number of semantic metadata and matchmaking. Such a task can be distributed to its Cloud utility services. Additionally, an MSNP peer can also synchronise some data to its private Cloud, possibly in the form of cached service description metadata documents.
- AMSNP Host — It represents an MSNP peer with embedded AMSNP framework. An AMSNP host is built based on ESB architecture. Each component of AMSNP is a service, and can be launched/terminated at runtime. A function can be performed by a local service within the AMSNP host, or it can be performed by an external service such as a private Cloud utility service

depends on the definitions of corresponding workflow pattern. AMSNP system is controlled by the WS-BPEL [21] workflow engine. When the user's application submits a request to AMSNP, the request will be handled by the *Request Handling* component, and a corresponding workflow pattern will be selected. The selected workflow pattern will then be passed to the workflow engine for execution via the message routing control component. Each workflow task is managed by a *Task Agent*. The *Task Agent* will decide how to perform the task after analysing the cost-performance scheme, which is described in the next section.

The AMSNP host contains the following main components:

- ◇ *Resource State Management* service is responsible for continually monitoring the resource usages such as CPU usage, network bandwidth usage, Cloud utility service usage, etc. These resource usages are cost intensive, and are the main elements influencing the decision making of the adaptation scheme in the next section.
- ◇ *Service Pool* is responsible for managing information on internal services, private Cloud services, and services provided by external MSNP peers. It contains a collection of the service descriptions of external MSNP peers, the service descriptions of each internal service and each accessible private Cloud utility service.
- ◇ *Functional Components* are miscellaneous utility components such as semantic metadata matchmaking component, calculation component (for calculating the CPI value in next section), message parsing, and so on.
- ◇ *Trust/QoS* and *Privacy/Security* are additional components needed to improve the quality of service and security requirements. They are not within the scope of this paper. We will consider them in our future work.

2.3 Adaptive Approach Selection Based on CPI Model

As we mentioned in the previous section, each request received by the *Request Handler*, is to be processed by triggering a corresponding business process workflow pattern. In a basic workflow pattern document (e.g., WS-BPEL), the endpoint (either a single service or a composite service) for processing each task/activity has been pre-defined in the document. Considering the dynamic nature of mobile P2P environment, the pre-defined endpoint may not be the best selection for the task. For example, a workflow is launched when the network has only 10 or less peers in existence. The workflow defines that the task for service discovery will be fully performed by a local host service of the device without using external distributed services. However, once the workflow is launched, the situation can change, there can be 50 more peers suddenly joining the network. Such a change can make the pre-defined approach no longer feasible. On the other hand, distributing tasks to external service (such as a service deployed on GAE) is not always the best approach because in many cases, performing tasks in local host is more efficient. This concern leads us to apply the dynamic adaptation technique, which is capable of identifying the best approach for each workflow task at runtime.

In this section, we propose an adaptation scheme that can decide which approach should be chosen for each workflow task at runtime based on the latency (timespan) of the approach, and costs. In order to clarify the terminologies used in this scheme, we first provide following definitions:

Definition 1: ***Approach* — *A*.** $A = \{a_j \in A : 1 \leq j \leq N\}$. Each $a_j \in A$ consists of a performance value (p), and a set of cost element values (E). Where $E^{a_j} = \{e_k^{a_j} \in E^{a_j} : 1 \leq k \leq N\}$.
The approach for a task is selected at runtime after the workflow is launched, and the decision is made based on the cost and performance.

Definition 2: ***Workflow pattern.*** A workflow pattern defines a goal and a set of sequential or parallel abstract tasks — T, where $T = \{t_i \in T : 1 \leq i \leq N\}$. Each $t_i \in T$ can be completed by numerous pre-defined approaches.
For example, a set of services — S ($S = \{s_i \in S : 1 \leq i \leq N\}$) has been discovered that can provide the content requested. The task of invoking each $s_i \in S$ to retrieve content, can be either performed by approach — a_1: using a localhost component to retrieve all content, or it can be performed by approach — a_2: distribute the process to a cloud service and then synchronise the result to user's mobile device.

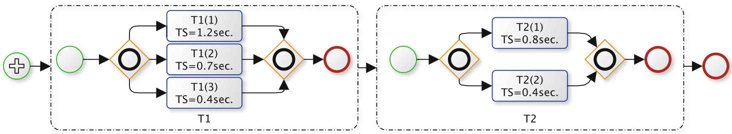


Fig. 3. Workflow path selection based on timespan

Fig. 3 shows a sample workflow which has two tasks. For task $T1$, there are three selective approaches, and for task $T2$, there are two selective approaches. Each approach consumes different timespan. In order to achieve the goal effectively, the system needs to identify the shortest path to reach the goal. Initially, the shortest path can be obtained by (1).

$$path_x = min \left\{ \sum_{i \in T, j \in A_i} p_j^i \right\} \tag{1}$$

Where p_j^i denotes the timespan of approach j of task i.

However, the shortest timespan may not mean the approach selection is the most efficient when the cost is considered. Hence, we propose a *cost-performance index* (CPI) scheme to enable our workflow system to analyse and select the most efficient approach at runtime. The scheme combines fuzzy set [38] and the *weight of context* [11]. The reason to choose fuzzy set approach is because the explicit purpose is to compare the performance and cost between approaches instead of static values. Hence, fuzzy set appeared to be a feasible solution.

Let D^{t_x} be a set of timespan value for the selective approaches (A_{t_x}) of task — t_x, where D is a finite set, and $D = \{d_i \in D : 1 \leq i \leq |A|\}$, in which d_i represents the timespan of a_i, $a_i \in A$. Let L be the longest timespan in D, where $L = max\{d_i \in D\}$. The performance value of each approach — R_i is computed by (2):

$$R_i = \begin{cases} 1 & \text{iff } d_i \equiv L \\ (L+1) - d_i & \text{otherwise} \end{cases} \tag{2}$$

Let $\tilde{A}$ be the fuzzy set of A. $\tilde{A} = \{\tilde{a}_j \in \tilde{A} : 1 \leq j \leq |A|\}$. We need the normalised fuzzy number of the ranking values. Hence, the fuzzy number of an approach's ranking value (denoted by $\tilde{a}_x$) will be: $\tilde{a}_x = R_x / \sum_{a_j \in A} R_j$. Where R_x is the performance value of a_x derived from (2), and $\tilde{a}_x$ is the normalised fuzzy number of the performance value of a_x, in which $0 \leq \tilde{a}_x \leq 1$.

At this stage, we assume there is a mechanism that can measure the timespan for each approach at runtime based on our previous work [8]

Definition 3: *Cost element* — E^{a_j} is a finite set, where $E^{a_j} = \{e_k \in E^{a_j} : 1 \leq k \leq N\}$. An a_j contains an E^{a_j}, and the value of e_k is denoted by v_{e_k}.

The cost element set is comparable between different related approaches. If approach a_1 for task t_1 — $E^{t_1}_{a_1}$ contains the value of "battery cost", then the approach a_2 for task t_1 — $E^{t_1}_{a_2}$ must also contain such a value. Based on this concept, the overall CPI between different approaches can be compared.

Since we are comparing the cost element between different approaches, the normalised value of a cost element — $\tilde{v}_{e_x}$ can be computed from $\tilde{v}_{e_x} = \frac{v_{e_x}}{\sum_{e_k \in E} v_{e_k}}$, and the average value of the total cost of a_j — $CV^{t_i}_{a_j}$ can be computed from $CV^{t_i}_{a_j} = \frac{\sum_{e_k \in E_{a_j}} \tilde{v}_{e_k}}{|E_{a_j}|}$. By applying the basic CPI model, the cost-performance value — δ of an approach — a_j will be:

$$\delta^{t_i}_{a_j} = \frac{\tilde{a}_j}{CV^{t_i}_{a_j}} \tag{3}$$

However, the importance of weight of an e_k is different for different users. For example, when the device battery-life remains 50%, the user may consider that saving the battery life of his/her mobile device is more important than spending money on using Cloud services for computational needs. In this case, the weight of the battery life cost element will be higher than the weight of the bandwidth cost of the Cloud service. Therefore, the normalised value of an e_k needs be refined as $\tilde{v}_{e_k} \cdot w_{e_k}$, where w_{e_k} denotes the weight of e_k, and the cost will be refined as follow:

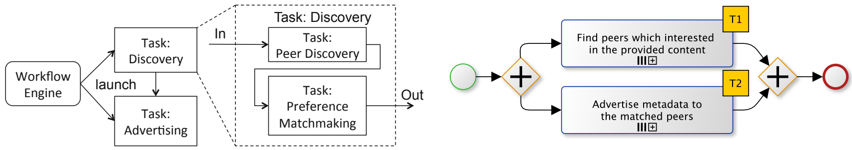
$$\hat{C}^{t_i}_{a_j} = \frac{\sum_{e_k \in E} \tilde{v}_{e_k} \cdot w_{e_k}}{\sum_{e_k \in E} w_{e_k}}, w_{e_k} \geq 1 \tag{4}$$

Finally, the cost-performance value of a_j will be refined as:

$$\delta^{t_i}_{a_j} = \frac{\tilde{a}_j}{\hat{C}^{ti}_{aj}} \tag{5}$$

3 Example

In this section, we use an example to show how the workflow system can be applied to an MSNP scenario. In the scenario, a MSNP peer (*PeerX*) intends to advertise content recommendation metadata (*CRM*; describing the URIs of the recommended content/service) to other MSNP peers. Fig. 4(a) illustrates the conceptual workflow of the content advertising process, and Fig. 4(b) describes the workflow in Business Process Modelling Notation (BPMN) [34]. BPMN has been chosen to describe the workflow process because it can be mapped to WS-BPEL [23], and WS-BPEL has been used in our prototype to control the processes. In this example, the workflow consists of two parallel tasks operated asynchronously (see Fig. 4(b)):



(a) Conceptual workflow **(b)** Content advertising

Fig. 4. Content advertising workflow

- *discovery* (T1) — discovers peers which are interested in the content. T1 consists of two sub-tasks: *Peer Discovery* and *Preference Matchmaking*. *Peer Discovery* denotes the process of discovering physical peers in MSNP environment and retrieving the content/service preference metadata from the peer. The result of *Peer Discovery* will be sent to *Preference Matchmaking* for determining whether the peer is interested in the provider's content/service or not. The result of *Preference Matchmaking* will be represented as the result of T1, and will be sent to T2.
- *advertising* (T2) — sends *CRM* to the matched peers.

Each task is managed by a task agent and the basic task handling workflow is described in Fig. 5. When a task is launched, the first step (S1 in Fig. 5) defines a feasible approach based on the CPI scheme described in the previous section. In step 2 (S2), an *event gateway* is placed. The task agent will enter the standby

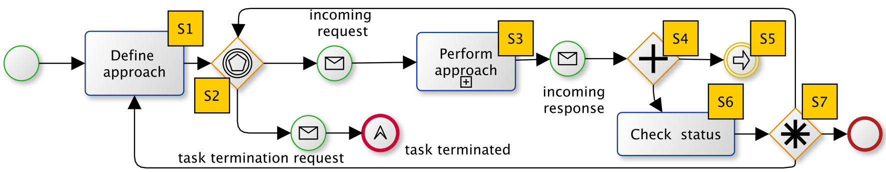
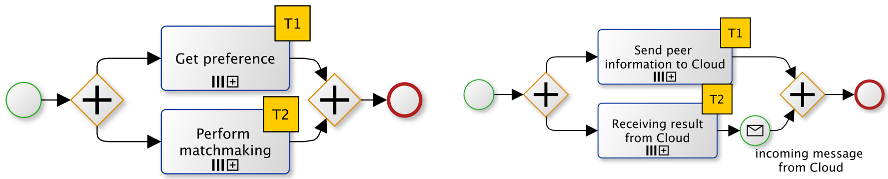


Fig. 5. Generic task



(a) Mobile-based discovery **(b)** Cloud-based discovery

Fig. 6. Approaches

mode to receive the incoming messages. There are two kinds of requests sent to the task agent: (1) the general request for the task; or (2) the termination request, which informs the task agent to terminate its task state. If the task involves activating a localhost service, when the termination request is received, the task agent will terminate the launched service, and inform the workflow engine to release the task agent from memory.

If the task agent receives an incoming request, it will perform the selected approach (S3). In this example, two approaches have been defined for task 1 (see T1 in Fig. 4), which are mobile-based discovery (Fig. 6(a)) and cloud-based discovery (Fig. 6(b)). Each is a sub-workflow and consists of two parallel tasks. For the approach in Fig. 6(a), the task agent will perform a sub-process (Fig. 6(a) — T1) to retrieve the service preference metadata from each MSNP peer in the network. The response message received by Fig. 6(a) — T1 will be passed to Fig. 6(a) — T2 for service matchmaking process. As for Fig. 6(b), which is the cloud-based approach, the mobile host will send a request to its Cloud utility service (CloudUtil) when an MSNP peer is found (see Fig. 6(b) — T1). The request message contains the basic information about the peer (e.g, the URL to retrieve its current IP address), the CloudUtil will retrieve and process the service preference metadata from each MSNP peer to find out which peer is interested in the content provided by the *PeerX*. The parallel task (Fig. 6(b) — T2) is launched at the same time as Fig. 6(b) — T1 to receive result from the CloudUtil.

The result of Fig. 6(a) — T2 or Fig. 6(b) — T2 will be sent to the original workflow (see "incoming response" in Fig. 5). When the original workflow receives the response, it reaches the *parallel gateway* (see S4 in Fig. 5) in which two activities will be performed. The first is to forward the response message to the next task (S5). In this example, the result from service matchmaking will be sent to the task agent which manages the advertisement task (Fig. 4 — T2). The second activity is to check the status by interacting with the *Resource State Management* component (see Section 2.2). The status check activity can result three possible condition:

– If the current status has changed (e.g., a large number of peers have joined the network, or the device battery life has reached a specific level), the task agent will perform the "define approach" step again.

– If the task is completed (e.g., the task has been defined that the advertisement will be only pushed to 50 peers and there are 50 peers which have been discovered), the task will be terminated.
– if the previous two conditions were not met, the task state will remain, and the task agent will continue to perform the same approach when it receives the incoming request.

4 Prototype

A prototype has been implemented using Objective-C and has been installed and tested in an *iPod Touch 4th generation* [3]. Currently, the prototype's workflow process component can process $< sequence >$ and $< flow >$ of WS-BPEL 2.0 documents. We simulate the other MSNP peers by deploying different number of Web service provider peers in a Macbook (2008), and each peer is published in a Bonjour network as a Web service provider. In this test case, each peer has a back-end Cloud storage using Dropbox, and the peer's current IP address is continually synchronising to its Cloud storage, and is retrievable from a static URL address using HTTP GET request. Moreover, since each peer is a Web service provider, the communication does not rely on the common Web service request/response process. Instead, when two peers initiate the communication, they exchange their basic description metadata, which contains the information of which URL provides the peer's current IP address. By doing so, a requester does not wait for the response when it send out the query, instead the request query contains a specific ID. When the provider complete the request, it invokes the requester node and sends the result (with the specific ID contained in the requester's query) to the requester.

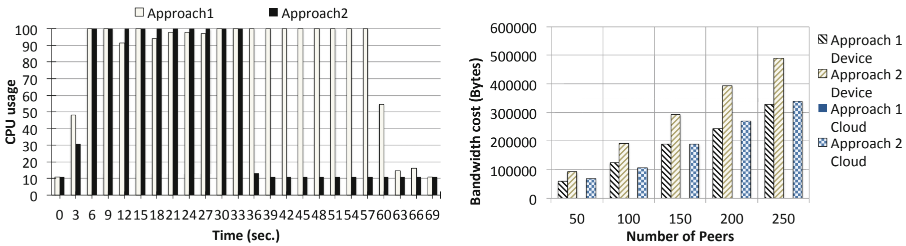
4.1 Evaluation

The evaluation aims to show how the adaptation mechanism changes its approach at runtime based on the cost and performance. The test case was based on the scenario described in Section 3 previously.

At the start of the test case, 10 MSNP peers have been found. After the workflow is executed, more peers have joined the network. Hence, the system needs to perform the calculation to identify whether the approach should change or not, based on the CPI values of approaches.

In the experiment, three cost elements have been considered: CPU usage of mobile device, network bandwidth cost of mobile device, network bandwidth of the Cloud utility service. In a networked system, CPU usage and network transaction costs are two of the main elements that consume the most battery-life of a mobile device. The Cloud bandwidth cost has been considered because it is one of the limitations of GAE. Note that the cost element of the Cloud in this evaluation was only used to show how the system behaves based on the proposed CPI scheme. In reality, the cost of a Cloud utility service such as the application that has been deployed on GAE or Amazon EC2 can involve other factors such as instance creating platform, hardware performance, time of usage, etc.

Mobile devices have limited processing capacity. In the test, tasks were performed asynchronously. Our experiment involved 250 MSNP peers and the total cost of using GAE is within its free usage plan limitation. If there were more than 250 MSNP peers involved, the device is unable to handle its tasks efficiently within an acceptable timespan. Hence, we did not consider the pure cost elements of Cloud like those in Amazon EC2. In the following discussion, Approach 1 represents a workflow consisting of T1A1, and T2; Approach 2 represents a workflow consisting of T1A2, and T2. T1 and T2 are parallel tasks and their sessions will remain until the workflow is terminated. For example, the entire process can be set for a specific period, and it will terminate when the period expired.

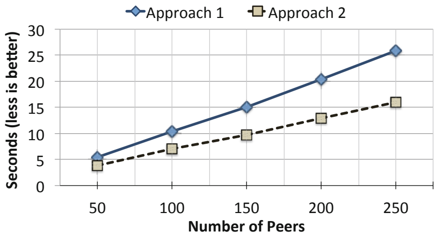


(a) CPU usage **(b)** Bandwidth cost

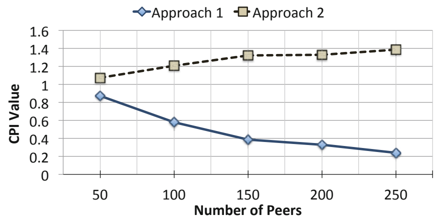
Fig. 7. Cost records

Fig. 7(a) illustrates the recorded CPU usage of the two approaches. The figure shows that while the application is running, it consumes around 11% of CPU usage. This is because the device is running a Web server and has joined Bonjour network, in which the device needs to continually communicate with the router to update the Bonjour service list. At the 3 second mark, the workflow has been triggered, so the CPU usage goes to 100%. For Approach 1, the CPU usage over 90% for 51 seconds. On the other hand, for Approach 2, the CPU usage over 90% was 27 seconds. The CPU usage cost element of our experiment was based on how long the CPU usage stays at over 90%. In Fig. 7(a), Approach 1 costs 24 more seconds than Approach 2. Fig. 7(b) illustrates the bandwidth cost recorded for both device-side and the Cloud utility service-side for different member of MSNP peers in the network. 'Device' denotes the bandwidth cost of the *MCP*'s device. 'Cloud' denotes the bandwidth cost of the Cloud utility service. Since Approach 1 does not use Cloud utility service, the cost value of 'Approach 1 Cloud' is always zero.

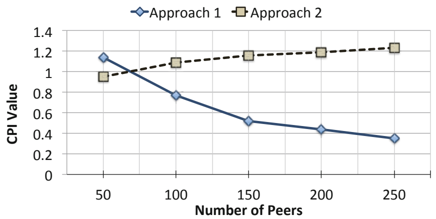
Fig. 8(a) illustrates the process timespan recorded for each approach influenced by the number of MSNP peers. As the figure shows, with fewer the number of peers, Approach 2 (which distributes the matchmaking process to Cloud) does not improve the performance much. Fig. 8(b) illustrates the CPI values of both approaches influenced by the number of peers. In this case, the weight of each cost element has been set equally to 1. As the number of peers increases,



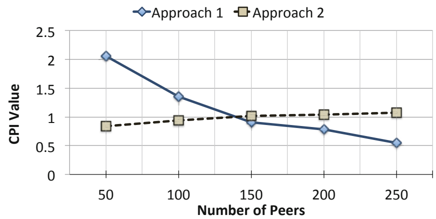
(a) Timespan (lower is better)



(b) CPI values with equal weight of cost element (higher is better)



(c) Cloud bandwidth weight + 1 (higher is better)



(d) Cloud bandwidth weight + 5 (higher is better)

Fig. 8. Cost performance index testing result

the CPI value of Approach 1 is reduce. In the next case, we assume the system intends to reduce the Cloud bandwidth usage because the available bandwidth deserving the free-of-charge period is getting low. Hence, the weight of Cloud bandwidth is increased by 1. The result (Fig. 8(c)) shows that when the number of peers is 50, the CPI value of Approach 1 is higher than Approach 2. Hence, the workflow remains in Approach 1. In the final case (Fig. 8(d)), we assume that the available bandwidth of free-of-charge period is nearly reach the end. Hence, the weight of the Cloud bandwidth is increased by 5. In this case, the workflow engine only selects Approach 2 when there are 150 peers or more.

5 Related Works

In recent years, a number of works have been proposed to enable proximal-based MP2P social network. However, existing decentralised MSNs are still in their early stages. Works such as [37,25,36] were focused on how to enable the OSN activities in mobile P2P networks. Within decentralised MSN, two works have focused on how content can be shared. The authors in [1] have modelled the user's interest profiles, and also introduced a formal mathematical scheme to decide how the content can be proactively pushed to the friends/contacts with potential interest in the content. The authors in [16] have proposed ontology-based formal semantic models to enable content sharing using semantic content matchmaking scheme. The approach enables the user-interests-based content routing in decentralised MSN by analysing the similarity of user profiles. A common limitation in existing decentralised MSN solutions is that they were tightly-coupled solutions with limited flexibility and scalability. The AMSNP framework proposed in this paper is a service-oriented solution based on ESB architecture design and standard technologies, which allows fundamental resources used in the participants' interaction to be changed dynamically at runtime.

Workflow management systems (WfMS) enable autonomous processes, which can highly reduce user's interference in content mashup and content advertisement scenarios. Researchers [18,20] in MP2P area usually apply WfMS in specialised purpose scenarios such as field-work, rescue operations or disaster events, in which the involved mobile nodes are manageable, and collaborate for the same goal. Workflow adaptation schemes in these works focused on failure recovery or resource allocation. This is understandable because MP2P systems (in particular: mobile ad hoc network — MANET) deal with special purpose scenarios rather than general-purpose scenario [10] like in MSNP. Few works have been done on proposing workflow systems for MP2P content mashup. In [24], the authors have proposed a workflow system based on a Java API — AmbientTalk for mashup in MP2P environment. The work mainly focused on how to implement workflow tasks on-top of AmbientTalk. In [5], an adaptive workflow scheduling scheme has been presented for mobile ad hoc network in disaster scenario. These works have been designed for similar MP2P environments such as MSNP. However, they did not address issues raised in this paper. In this paper, the adaptivity of workflow mainly focuses on how to select the most feasible approach to complete the task of content mashup process based on performance (e.g., timespan of the approach) and costs (bandwidth, battery, transaction-load etc.).

6 Conclusion and Future Work

In this paper, we have proposed a workflow-based adaptive mediation framework for service-oriented MSNP. The framework enables a MSNP participating device to dynamically change its behaviour to adapt to different situations when it receives a user's request. The adaptation mechanism utilises the proposed CPI scheme to support the device to automatically select a feasible approach for each task within a request handling process by comparing the dynamically changed cost and performance of the approaches.

Workflow systems provide flexibility and scalability of MSNP processes. The adaptation scheme introduced in this paper enables the system to select a feasible approach to complete the workflow task. It also potentially brings a new form of MSNP communication. For example, an active peer in a MSNP environment can provide a recommended routing approach (described in WS-BPEL) to a new peer joining the network. The new peer can automatically execute the WS-BPEL workflow process to perform service discovery or content retrieval without the need for user's manual control.

In the future, we will model different types of mobile P2P communication protocols in WS-BPEL and develop a more advanced MSNP environment simulator to evaluate our framework. Moreover, we intend to distribute more workflow tasks to different Cloud services to compare the cost and performance of different MSNP approaches.

References

1. Allen, S.M., Colombo, G., Whitaker, R.M.: Uttering: social micro-blogging without the internet. In: The 2nd Int. Workshop on Mobile Opportunistic Networking, pp. 58–64. ACM (2010)
2. Amazon: Amazon Elastic Compute Cloud, http://aws.amazon.com/ec2/
3. Apple: iPod Touch, http://www.apple.com/ipodtouch/
4. Apple: Open Source: Bonjour, https://developer.apple.com/opensource/
5. Avanes, A., Freytag, J.C.: Adaptive workflow scheduling under resource allocation constraints and network dynamics. Proc. VLDB Endow. 1(2), 1631–1637 (2008)
6. Bluetooth: Bluetooth, http://www.bluetooth.com/Pages/Bluetooth-Home.aspx
7. Chang, C., Ling, S., Krishnaswamy, S.: Promws: Proactive mobile web service provision using context-awareness. In: PerCOM Workshops 2011, pp. 69–74 (2011)
8. Chang, C., Srirama, S.N., Krishnaswamy, S., Ling, S.: Proactive web service discovery for mobile social network in proximity. MENERVA, JNIT (to be published, 2012), http://dl.dropbox.com/u/59860036/service_discovery.pdf
9. Comiket: Official Comic Market Site, http://www.comiket.co.jp/index_e.html
10. Conti, M., Giordano, S.: Multihop ad hoc networking: The reality. IEEE Communications Magazine 45(4), 88–95 (2007)
11. Delir Haghighi, P., Krishnaswamy, S., Zaslavsky, A., Gaber, M.M.: Reasoning about Context in Uncertain Pervasive Computing Environments. In: Roggen, D., Lombriser, C., Tröster, G., Kortuem, G., Havinga, P. (eds.) EuroSSC 2008. LNCS, vol. 5279, pp. 112–125. Springer, Heidelberg (2008)
12. Dropbox: Dropbox, http://www.dropbox.com/
13. Facebook: Facebook, http://www.facebook.com/
14. Google Inc.: Google App Engine, https://developers.google.com/appengine/
15. IETF: Zero Configuration Networking (Zeroconf), http://www.zeroconf.org/
16. Li, J., Wang, H., Khan, S.: A semantics-based approach to large-scale mobile social networking. In: Mobile Networks and Applications, pp. 1–14 (2011)
17. Lubke, R., Schuster, D., Schill, A.: Mobilisgroups: Location-based group formation in mobile social networks. In: 2011 IEEE Int. Conf. on Pervasive Computing and Communications Workshops, pp. 502–507 (March 2011)
18. Mecella, M., Angelaccio, M., Krek, A., Catarci, T., Buttarazzi, B., Dustdar, S.: Workpad: an adaptive peer-to-peer software infrastructure for supporting collaborative work of human operators in emergency/disaster scenarios. In: Int. Symposium on Collaborative Technologies and Systems, pp. 173–180 (May 2006)
19. Merrill, D.: Mashups: The new breed of web app. IBM developerWorks (2006), http://public.dhe.ibm.com/software/dw/xml/x-mashups-pdf.pdf
20. Neyem, A., Franco, D., Ochoa, S.F., Pino, J.A.: An Approach to Enable Workflow in Mobile Work Scenarios. In: Shen, W., Yong, J., Yang, Y., Barthès, J.-P.A., Luo, J. (eds.) CSCWD 2007. LNCS, vol. 5236, pp. 498–509. Springer, Heidelberg (2008)
21. OASIS: WS-BPEL 2.0 (2007), http://docs.oasis-open.org/wsbpel/2.0/wsbpel-v2.0.html
22. OASIS: DPWS (2009), http://docs.oasis-open.org/ws-dd/ns/dpws/2009/01

23. Ouyang, C., Dumas, M., ter Hofstede, A.H.M., van der Aalst, W.M.P.: From bpmn process models to bpel web services. In: ICWS 2006, pp. 285–292 (2006)
24. Philips, E., Carreton, A.L., Joncheere, N., De Meuter, W., Jonckers, V.: Orchestrating nomadic mashups using workflows. In: The 3rd and 4th Int. Workshop on Web APIs and Services Mashups, pp. 1:1–1:7. ACM (2010)
25. Pietiläinen, A.K., Oliver, E., LeBrun, J., Varghese, G., Diot, C.: Mobiclique: middleware for mobile social networking. In: The 2nd ACM Workshop on Online Social Networks, pp. 49–54 (2009)
26. Robinson, R.: Understand Enterprise Service Bus scenarios and solutions in Service-Oriented Architecture, Part 1, http://www.ibm.com/developerworks/webservices/library/ws-esbscen/
27. Sapuppo, A.: Spiderweb: A social mobile network. In: The 2010 European Wireless Conference, pp. 475–481. IEEE (2010)
28. Schuster, D., Springer, T., Schill, A.: Service-based development of mobile real-time collaboration applications for social networks. In: The 8th IEEE Int. Conf. on Pervasive Computing and Communications Workshops, pp. 232–237. IEEE (2010)
29. Srirama, S., Jarke, M., Prinz, W.: Mobile web service provisioning. In: Int. Conf. on Internet and Web Applications and Services, pp. 120–125 (2006)
30. The XMPP Standards Foundation: XMPP, http://xmpp.org/
31. Tsai, F.S., Han, W., Xu, J., Chua, H.C.: Design and development of a mobile peer-to-peer social networking application. Expert Syst. Appl. 36, 11077–11087 (2009)
32. twitter: twitter, http://www.twitter.com/
33. UPnP Forum: Universal Plug and Play, http://www.upnp.org/
34. White, S.A.: Introduction to BPMN, http://www.omg.org/bpmn/Documents/Introduction_to_BPMN.pdf
35. Wi-Fi Alliance: Wi-Fi Direct™, http://www.wi-fi.org/discover-and-learn/wi-fi-direct%E2%84%A2
36. Xing, B., Seada, K., Venkatasubramanian, N.: Proximiter: Enabling mobile proximity-based content sharing on portable devices. In: PerCOM 2009, pp. 1–3. IEEE Computer Society (2009)
37. Yang, G., Liu, Z., Seada, K., Pang, H.Y., Joki, A., Yang, J., Rosner, D., Anand, M., Boda, P.P.: Social proximity networks on cruise ships. In: MIRW, pp. 105–114. ACM (2008)
38. Zadeh, L.: Fuzzy sets. Information and Control 8(3), 338–353 (1965), http://www.sciencedirect.com/science/article/pii/S001999586590241X