A Cross-Layer Design for Adaptive Multimodal Interfaces in Pervasive Computing

Jun Kong Weiyi Zhang Juan Li Arjun G. Roy

Department of Computer Science
North Dakota State University
{jun.kong, weiyi.zhang, j.li, arjun.roy}@ndsu.edu

ABSTRACT

Multimodal interfaces have attracted more and more attention. Most researches focus on each communication mode independently and then fuse the information at the application level. Recently, several frameworks and models have been proposed to support the design and development of multimodal interfaces. However, it is still a challenging issue of supporting adaptations in the multimodal interfaces. Existing approaches are using rule-based specifications to define the adaptation of input/output modalities. Distinct from previous work, this paper presents a novel approach, which quantifies the usability of each modality and then formalizes the adaptation issue as searching for a set of input/output modalities that produce the highest usability. Furthermore, our approach supports a cross-layer design, which considers the adaptation from the perspectives of the interaction context, available system resources and QoS requirements. In other words, our design crosses application, system and network layers. An optimal solution and a heuristic algorithm are developed to automatically select an appropriate set of modalities combinations under a certain situation. The numerical evaluation shows good performance of both solutions.

Categories and Subject Descriptors

D.2.2 [Design Tools and Techniques]: User interfaces

General Terms

Human Factors, Design

Keywords

Multimodal interfaces, adaptation, linear programming

1. INTRODUCTION

Multimodal interfaces process two or more combined user input modes (such as speech, pen, gaze, or manual gestures) in a coordinated manner with multimedia system outputs [Ovi02]. With a growing consensus on improved performance of multimodal interaction, there have been many advances in multimodal HCI. Currently, multimodal interfaces are being commercialized for in-vehicle, smart phone and other applications, as illustrated by multimodal interfaces created by SpeechWorks and Ford at the 2003 North American International Auto Show [Ovi04]. A majority of research approaches focus on each communication mode independently and then fuse the information at the application level [Jai05]. Several frameworks and models [Bou04, Dra04, Fli03] have been proposed to support the design and development of multimodal interfaces.

One main challenge in multimodal interaction design is to automatically select an optimal set of modality combinations that users will find easy and intuitive to produce and that the system will be able to interpret [Bou07]. In a pervasive environment, modality combinations should further adapt to different interaction contexts, such as physical environments, due to the user mobility. In other words, mobile users may be moving around different locations with a handhold device while accessing information, which can cause a dynamically changing interaction context. Accordingly, the high dynamics requires an adaptive multimodal interface, which can provide high usability under a certain interaction context. However, few studies have been conducted on adapting a multimodal interface to different interaction contexts. The frameworks of FAME [Dua06] and MOSTe [Rou05] are valuable for developing adaptive multimodal applications. However, those approaches use rule-based specifications to define adaptation, which has the problems of completeness and coherence [Rou05]. This paper proposes a novel approach, which considers adaptation from three layers: the interaction context in the application layer, the resources allocation in the system layer and the QoS provisioning in the network layer.

In human computer interaction, computers and humans establish various communication channels, over which messages are exchanged with associated effects [Obr07]. However, user perceived effects may be reduced by constraints on interaction platforms, physical environments and the static and dynamic features of a user. For example, visual effect may be limited by a small screen on a mobile device; a noisy environment can greatly reduce the usage of auditory effects; and blind users are absent of all visual stimulus processing. Furthermore, user's dynamic features, such as walking and driving, can also influence interaction effects. For example, a fast movement (i.e. running) can affect a user reading information and selecting a menu in an interaction. Behavioral rules [Dua06] have been commonly used to specify the adaptation, triggered by a change in the interaction context. Distinct from existing work, our approach quantifies the usability of each modality. Based on the quantification, the issue

Conference'04, Month 1–2, 2004, City, State, Country.

of modality adaption is formalized as an optimization issue with the objective of maximizing the usability under an interaction context with certain constraints satisfied.

In addition to dynamic interaction contexts, handheld devices in a pervasive environment in general are constrained with limited computing capability, which raises an interesting topic of applying multimodal HCI to mobile devices that have limited input/output resources [Jai05]. On a mobile device, different modalities compete for limited system resources, such as battery and CPU. For example, in an education application, PowerPoint slides are associated with a video of the instructor's lecture. However, the PPT may not be displayed simultaneously with the video due to limited CPU and network capacity. Therefore, we must assure that the resources requested by selected modalities cannot exceed the total available resources.

Different modalities may have different QoS requirements. For example, an auditory message requires higher bandwidth for a short delay than a textual message. So, it requires a QoS-assured path for one selected modality. QoS-assured routing is a fundamental issue in wireless and wired networks, and has been extensively investigated. This paper considers delay as the QoS parameter and uses the shortest path algorithm to search for a satisfied routing path. In the case of multiple QoS constraints, a randomized algorithm [Kor01] can be applied to solve the multi-constraint QoS path problem. Developing an algorithm for a multi-constraint QoS path is out of the scope of this paper.

In summary, this paper proposes a cross-layer design, which specifies adaptation from the application layer (i.e. interaction context of user, platform and environment), the system layer (i.e. resources) and the network layer (i.e. QoS requirements). By quantifying each modality with a usability value, we formalize the adaptation issue as searching for input/output modalities with the objective of achieving *the highest usability* under constraints (1) do not exceed the maximum available system resources and (2) QoS requirements are assured for selected modalities. Without considering QoS requirements, the adaptation can be reduced to the classic 0-1 knapsack problem, which is NP-complete though it has a pseudo-polynomial time solution [Gar90]. Having QoS in mind, the adaptation problem is obviously NP-complete. This paper first proposes an optimal solution based on integer linear programming. Due to the high complexity of the optimal solution in a large problem, this paper also proposes an efficient heuristic algorithm to solve the adaption issue based on the classic 0-1 knapsack problem. Briefly speaking, 0-1 knapsack is first used to select a set of modalities, which achieve the highest usability and do not exceed available system resources; then, a shortest path algorithm is used to filter out the modalities selected in the first step, which do not have a QoS-assured routing; finally, according to a decreasing order of the usability of each unselected modality, an unselected modality is selected if the remaining system resources can accommodate its request and it has a QoS-Assured routing. We provide the numerical results for both the heuristic algorithm and the optimal solution, and show that both algorithms have a good performance.

The paper is organized as follows. Section 2 presents an overview of our approach. Section 3 gives an optimal solution based on linear programming. Section 4 introduces a heuristic algorithm. Section 5 presents the numerical simulation results. Section 6 goes through a case study. Section 7 discusses related work, followed by conclusion and future work in Section 8.

2. A CROSS-LAYER DESIGN FOR DEVELOPING MULTIMODAL INTERFACES

Figure 1 shows a cross-layer design for adaptive multimodal interfaces. Under a certain interaction context, a mapping from a modality space to a usability space defines an *interaction context profile*, which indicates the usability of a modality under a given interaction context; a mapping from a modality space to QoS preferences defines a *QoS profile*, which indicates the QoS requirements of a modality; and a mapping from a modality space to system requirements defines a *resource profile*, which indicates the resource requirements of executing a modality. Based on the user profile, the QoS profile and the resource profile, an optimal solution and a heuristic algorithm are designed to determine input/output modalities, which achieve the highest usability under a given interaction context with satisfied system requirements and a QoS-assured path.

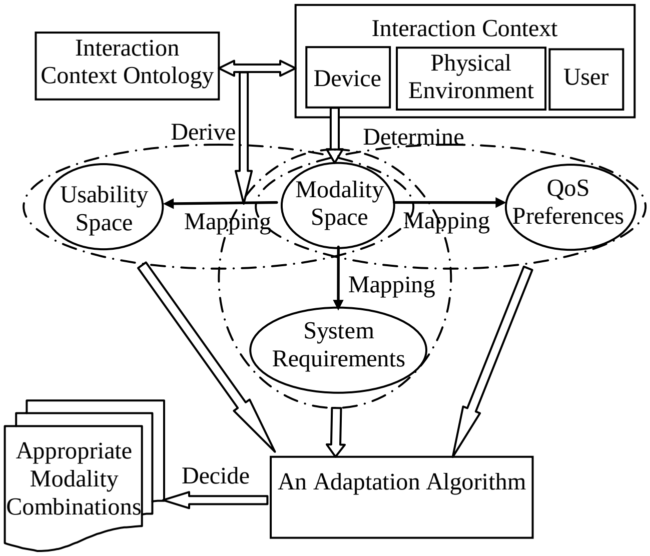

Figure 1. Design of Adaptive multimodal interfaces

2.1. Interaction Context

Interaction context describes the state under which a person uses a device [Abo98]. Getting inspired by previous researches [Pre04, Sch94, Sch99], we consider three main entities in the interaction context: *User*, *Device*, and *Environment*.

User plays a central role in the human computer interaction. The delivery and rendering of information to a user must fit user's personalized features, such as the user's preferences, motion, physiological and mental states, mood, current activity and etc. Input/output modalities must be adapting to users.

The **environment** in which the user interacts with the device is another important entity in the human computer interaction. The environment is not directly related to the user, but rather sensed through a device. It continuously affects the interaction of a communication mode. For example, a high noise level could significantly reduce an auditory effect. Important environmental concepts in this part of the context include: location, time, and environmental conditions, such as temperature and lighting.

The **device** entity determines the input and output capacities, i.e. a *modality space* that defines all available communication modes supported by the device. Furthermore, the characteristics of a device may affect the usability of a communication mode.

2.2. Modality Space

A device in a pervasive environment determines available input and output modalities, which construct a modality space. Under a given interaction context, a subset of appropriate modalities should be selected from the modality space. Since user mobility may change the infrastructure of a pervasive environment, a modality space needs to be updated accordingly.

The modalities in a modality space have two fundamental relations, i.e. redundancy and complementarity [Ovi99]:

- Redundancy: If two communication modes deliver the same amount of information, they have the redundancy relationship. For example, during computer-mediated, distance-learning lectures, 100% of the presenter's handwriting was accompanied by semantically redundant speech [And04a, And04b, Kai07].
- Complementary: The presentation and interpretation of information depend on two or more modalities. For example, speech and pen inputs consistently contribute different and complementary semantic information [Ovi97, Ovi99].

In a multimodal interface, the modalities with a redundancy relation can be selected independently. However, modalities with a complementarity relation must be selected at the same time, and they are considered as a unified composite modality in the process of modality adaptation. The execution of a modality needs to consume system resources. The consumption of system resources implicitly reflects the competition among modalities. Therefore, the modality adaption must consider the availability of system resources.

2.3. Usability space

Some researchers have investigated the suitability of a modality under different interaction contexts [Obr07, Lem08]. Based on previous work, the usability of a modality can be quantified as a value between 0 and 1. Such a usability value represents a user's personal satisfaction/preference on a modality. More specifically, a modality with value 1 indicates that a user can easily interact with a system through this modality. On the other hand, a modality with value 0 means that this modality is not usable for a user. For example, the output modality of photograph has a "0" usability for a blind user. The usability value of a modality is dynamic and may change according to an interaction context. For example, when a user is moving from his/her office to a vehicle and starts driving the vehicle, the usability value associated with a visual display will be significantly reduced. Under a certain interaction context, a mapping from a modality space to a usability space defines an interaction context profile, which represents the effect of a modality.

2.4. QoS Preferences

Information can be perceived by a user in different forms (such as text, image or speech), which correspond to different communication modes. Different forms may have different QoS requirements. For example, in order to support a smooth playback, speech-based information requires a higher bandwidth to have a short delay on a network than text-based information. To the best of our knowledge, most existing approaches do not consider QoS requirements. In a pervasive environment, network speed can change dramatically (e.g. a user may be connected from a wired high-speed LAN to a slow wireless network), which accordingly affects the usability of a communication mode. In this paper, we take *delay* as the primary QoS constraint, but our work can be easily extended to multiple QoS constraints.

2.5. System Requirements

The selection of a modality triggers the execution of a corresponding software application, which consumes system resources. For example, a speech-based interaction requires relevant voice recognition/synthesis software. Though the computing capability of a mobile device becomes more and more powerful, it is still constrained with physical limitations. Therefore, the modality adaption needs to consider the system requirements of each modality and assures that the requested resources do not exceed the maximum available resources. This paper considers three different types of system resources: CPU, memory and bandwidth while our work can easily extend to other resources.

2.6. Cross-Layer Multimodal Ontology Design

We designed an ontology [Gru95], i.e. CLMO (for Cross-Layer Multimodal Ontology), to define interaction context, system requirements and QoS requirements in an expressive and structured way. The CLMO ontology enables the sharing and reuse of the interface knowledge. It also supports various existing logic inference, which is vital to reason and compute the usability model. Our principle in ontology design is to create a general yet extendable ontology that will be able to describe different interaction situations when designing adaptive multimodal interfaces. The ontology definition includes three parts: the interaction context profile ontology, the QoS profile ontology, and the resource profile ontology. When defining the interaction profile ontology, we adopt some concepts from existing ontology such as GUMO [Hec05], CoOL[Str03], and DREAMS [Kor06]. Based on OWL [OWL] and RDF [RDF], Figure 2 shows the interaction context part of the CLMO ontology, which defines two classes "User" and "CellPhone" and two properties "isUsedBy" and "hasMotion". It also depicts an interaction scenario in which a particular user uses a particular cellular phone and is walking.

```
  <owl:Class rdf:about="#User">
      <rdfs:subClassOf rdf:resource="#Person"/> </owl:Class>

<owl:Class rdf:about="#CellPhone">
      <rdfs:subClassOf rdf:resource="#Device"/> </owl:Class>

<owl:ObjectProperty rdf:about="#isUsedBy">
      <rdf:type rdf:resource="&owl;InverseFunctionalProperty"/>
      <rdfs:domain rdf:resource="#Device"/>
      <rdfs:range rdf:resource="#User"/>
      <owl:inverseOf rdf:resource="#uses"/>
</owl:ObjectProperty>

 <owl:ObjectProperty rdf:about="#hasMotion">
      <rdfs:range rdf:resource="#Motion"/>
      <rdfs:domain rdf:resource="#Person"/>
</owl:ObjectProperty>
```

```
 ...

<User rdf:about="#user1">
      <rdf:type rdf:resource="&owl;Person"/>
      <uses rdf:resource="#blackberry_smartphone"/>
      <inEnvironment rdf:resource="#conference"/>
     ...
      <hasMontion rdf:resource="#walking"/> </User>

 <CellPhone rdf:about="#blackberry_smartphone">
      <rdf:type rdf:resource="&owl;Thing"/>
      <used_by rdf:resource="#user1"/>
             <has_CPU_capacity
rdf:datatype="&xsd;decimal">1G</has_CPU_capacity>
      <has_RAM_capacity
rdf:datatype="&xsd;decimal">32M</has_RAM_capacity>
      <has_bandwidhth_capacity
rdf:datatype="&xsd;decimal">56k</has_bandwidhth_capacity>
      <has_input_interface rdf:resource="#bluetooth_mouse"/>
      <has_output_interface rdf:resource="#phone_screen"/>
      <has_output_interface rdf:resource="#speaker"/>
            ...
 </CellPhone>
```

Figure 2. OWL/RDF description of a piece of the interaction context profile ontology

3. OPTIMAL SOLUTION

This section gives an optimal solution to the modality adaptation problem based on integer linear programming (ILP). The following notations have been used in the problem formulation:

- *MI*: The set of input modalities.
- *MO*: The set of output modalities.
- *MDS:* The set of available modalities, i.e. the *modality space*, and $MI \cup MO = MDS$.
- *BAND*: The total available system bandwidth of a pervasive computing system.
- *CPU*: The total available CPU capacity of the system.
- *MEM*: The total available memory of the system.
- *i*: Representing an input modality in the set of *MI.*
- *o*: Representing an output modality in the set of *MO.*
- *m*: Representing a modality in the set of *MDS.*
- *Usability(m)*: The usability value for the modality *m*.
- *Band*(*m*): The bandwidth required by the modality *m*.
- *Cpu(m)*: The CPU required by the modality *m*.
- *Mem(m)*: The memory required by modality *m*.
- *Delay*(*m*): Transmission *delay* constraint of the modality *m*.
- *Mod*(*m*): Indicating if a modality *m* is selected. The value 1 means that the modality is chosen. Otherwise the value is 0.
- *delay*(*i, j*): The delay of link (*i, j*) of the network.
- *Cap(i,j):* The capacity of link (*i, j*) of the network.
- *Adj(v)*: The adjacent nodes of node *v* in the network.

As defined in formula (1), the objective of modality adaptation is to *find a set of modalities, which achieve the maximum usability*. Furthermore, the selected modalities should include at least one input modality and one output modality, as defined in formulas (2) and (3). Formulas (4) to (6) define the constraints of system resources: requested resources of selected modalities should not exceed corresponding system resource capacities.

$$(1) Maximize\{\sum_{i\in MI} Mod(i) * Usability(i) + \sum_{o\in MO} Mod(o) * Usability(o)\}$$

$$(2) \sum_{i\in MI} Mod(i) \geq 1$$

$$(3) \sum_{o\in MO} Mod(o) \geq 1$$

$$(4) (\sum_{i\in MI} (Mod(i) * Band(i)) + \sum_{o\in MO} (Mod(o) * Band(o))) \leq BAND$$

$$(5) (\sum_{i\in MI} (Mod(i) * CPU(i)) + \sum_{o\in MO} (Mod(o) * Cpu(o))) \leq CPU$$

$$(6) (\sum_{i\in MI} (Mod(i) * Mem(i)) + \sum_{o\in MO} (Mod(o) * Mem(o))) \leq MEM$$

The corresponding software component of a modality has its delay constraint. In the following formulations from (7) to (11), s_m indicates the information receiver, in general the mobile user; and t_m represents the service provider. Formulas (7), (8), and (9) are used to find a path for a modality in the network. More specifically, $f^m_{(i,j)}$ indicates whether link (*i, j*) is used for routing modality *m*. The value 1 means that link (*i, j*) is on a path for modality *m*. Otherwise the value is 0. Formula (10) assures that the found path for modality *m* must satisfy the delay requirements of *m*. Formula (11) defines that each link in the network could be used for the transmissions of multiple applications, but the total traffic on the link cannot exceed its capacity.

$$(7) \sum_{i\in Adj(s_m)} f^m_{(s_m,i)} - \sum_{i\in Adj(s_m)} f^m_{(i,s_m)} = Mod(m), \forall m \in MDS$$

$$(8) \sum_{i\in Adj(t_m)} f^m_{(i,t_m)} - \sum_{i\in Adj(t_m)} f^m_{(t_m,i)} = Mod(m), \forall m \in MDS$$

$$(9) \sum_{i\in Adj(j)} f^m_{(j,i)} = \sum_{i\in Adj(j)} f^m_{(i,j)}, \forall m \in MDS, \forall i \in V \setminus \{s,t\}$$

$$(10) \sum_{(i,j)\in E} (delay((i,j)) * f^m_{(i.j)}) \leq Mod(m) * Delay(m), \forall m \in MDS$$

$$(11) \sum_{m\in MDS} (Band(m) * f^m_{(i.j)}) \leq Cap(i,j), \forall (i,j) \in E$$

The above linear programming formulas provide an optimal solution. When the network and the modality size are relatively small, it can return an optimal solution. However, when the problem input is large, solving this integer linear program could be time consuming. Next section introduces an efficient heuristic algorithm for the problem with a larger size.

```
Algorithm  Multimodal_Interface_Adaptation(User_Preferences; QoS; Resources)
1: sort the input modalities according to the usability values under current context model;
2: sort the output modalities according to the usability values under current context model;
3: Select an input modality i and an output modality o, which can result in the highest usability value and satisfy the resource constraint and QoS requirements;
4: Update the resource constraints;
5: Modality Set = all Available Modalities – Selected Input Modality i – Selected Output Modality o;
6: Use 0-1 Knapsack dynamic programming to find modalities, which produce the maximum usability;
7: Insert chosen modalities into set Chosen Knapsack;
8: Insert the unchosen modality into set Backup Knapsack;
9: Final Knapsack = Selected Input Modality + Selected Output Modality;
10: while (Chosen Knapsack is not empty) do
11:     Choose and delete the modality mod which has the maximum usability from the remaining modalities in the Chosen Knapsack;
12:     Shortest_Path(delay(mod));
13:     if (found a path for modality mod) and (the remaining system resource can satisfy mod) then
14:         Add mod into set Final Knapsack;
15:         Update the resources;
16:     end if
17: end while
18: Calculate the left available resource;
19: while (Backup Knapsack is not empty) do
20:     Choose and delete the modality mod which has the maximum usability in the Backup Knapsack;
21:     if (the remaining system resource can satisfy mod) then
22:         Shortest_Path(mod);
23:     if (found a path for mod) then
24:         Add mod into set Final Knapsack;
25:         Update the system resource;
26:     end if
27: end while
28: Return the modalities in the Final Knapsack ;
```

Figure 3. A heuristic algorithm

4. A HEURISTIC ALGORITHM

Briefly speaking, an efficient heuristic approach is developed to solve the modality adaptation in three steps:

- Search for a set of input and output modalities, which can achieve the maximum usability and do not exceed the system resource capacities. This step is essentially a 0-1 knapsack problem. After this step, all modalities are classified as selected and unselected.
- Then, we calculate the shortest path one by one for each modality m selected in the first step. If a path with satisfied delay constraint for modality m exists, m is finally selected. Otherwise, m is removed from the selected set.
- Starting from the modality with the highest usability in the *unselected* set, if a path with satisfied delay constraint for modality m exists, m is finally selected. This process continues until every modality in the unselected set is checked or no available resource left.

Figure 3 shows the proposed heuristic algorithm. In order to guarantee that selected modalities include at least one input modality and one output modality, lines 1-3 use a brute-force method to search for the composition of an input modality and an output modality, which have the highest usability with a QoS-assured routing path and satisfied resource requests. Then, among the remaining modalities, lines 4 to 6 use 0-1 knapsack dynamic programming to select a set of modalities with the highest usability and satisfied resource requests. Lines 7 to 17 check if a QoS assured routing exists for a modality selected in 0-1 knapsack. If such a path exists, the modality is finally selected. Otherwise, it is unselected for the modality adaptation. Finally, lines 18 to 27 check each modality m, which is unselected in the 0-1 knapsack, in a descending order based on its usability. If m has a QoS-assured routing and the remaining system resources can satisfy its request, m is finally selected in the modality adaptation. Otherwise, it is unselected.

5. NUMERICAL EVALUATION

In this section, we implemented our heuristic algorithm (denoted by **HEU** in the figures), and compared it with the optimal solution (denoted by **OPT** in the figures) given by the integer linear programming (ILP) formulation in Section 3. As in [Xue07], we used both well-known Internet topologies and randomly generated topology to study the suitability and computational time complexity of the algorithms. All tests were performed on a 2.4GHz Linux PC with 2G bytes of memory.

For each input or output modality, its usability was uniformly generated in a given range [0,1]. And its bandwidth, CPU and memory resource requirements were uniformly generated in a given range [1, R]. We consider different sets of modality resource requirements, **T**ight bound (R = 30), **A**verage bound (R = 50), and **L**oose bound (R = 70).

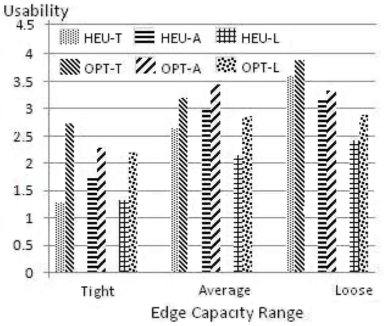

Figure 4. Usability on a random network with 2 input modalities and 8 output modalities

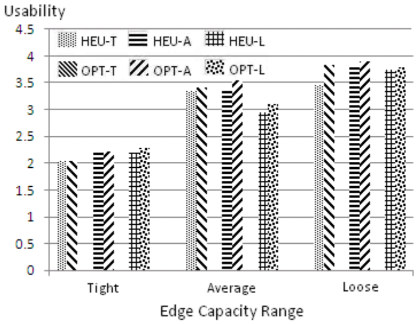

Figure 5. Usability on a random network with 4 input modalities and 6 output modalities

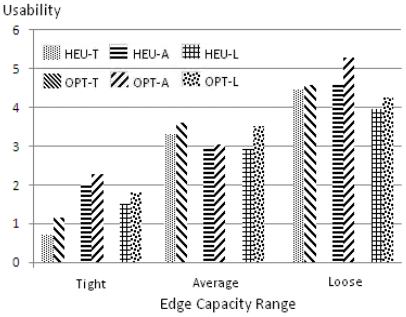

Figure 6. Usability on a random network with 5 input modalities and 5 output modalities

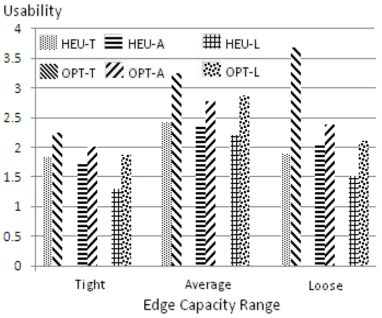

Figure 7. Usability on an NSF network with 2 input modalities and 8 output modalities

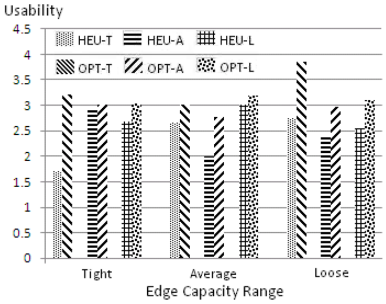

Figure 8. Usability on an NSF network with 4 input modalities and 6 output modalities

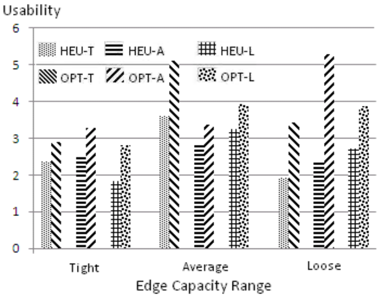

Figure 9. Usability on an NSF network with 5 input modalities and 5 output modalities

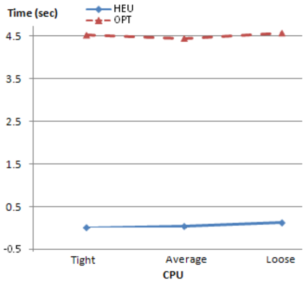

Figure 10. Running time on a random network with 2 input modalities and 8 output modalities

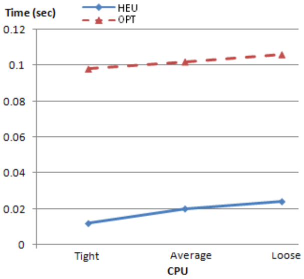

Figure 11. Running time on an NSF network with 5 input modalities and 5 output modalities

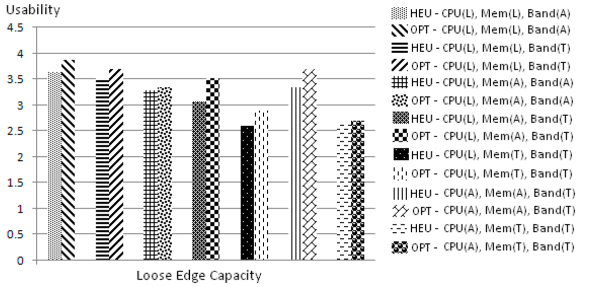

Figure 12. Usability on a random network with different system resources

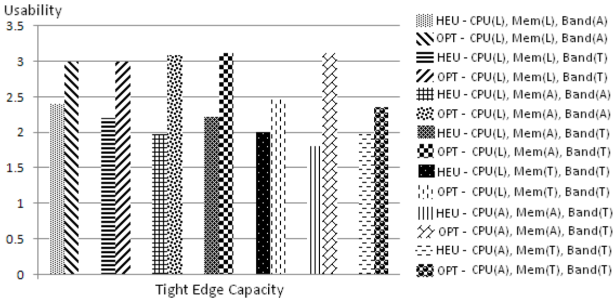

Figure 13. Usability on an NSF network with different system resources

In our simulation, different modality sets were generated. The size of the input/output modality sets are given as: *different* (2 input modality/8 output modality), *equal* (5 input modality/5 output modality) and *close* (4 input modality/6 output modality).

First we compare our solutions on a random generated network (50 nodes, 300 edges). Meanwhile, on each link in the network, as in [Kor01, Xue07], the delay of each link was also randomly generated in a given range (we used the range [1,10]). For each test case, we considered three scenarios according to the range of the capacity of the links: tight case (given range (0,10]), average case (given range (0,20]) and loose case (range (0,100]). In Figure 4, we show the usability performance on the random network with *different* modalities (2 input modalities and 8 output modalities). As expected, **OPT** always performed better than

HEU did regardless of modality resource requirements (**T**, **A** and **L** represents ***T**ight* bound, ***A**verage* bound, and ***L**oose* bound, respectively). Meanwhile, we can see that **HEU** delivered very good performance because it provided near-optimal solutions in most cases. Similar observations can be made for the *close* set of modalities in Figure 5, and for the *equal* set of modalities in Figure 6.

Same experiments had been tested on a well-known network topology NSFNET (14 nodes, 42 edges). The results are shown in Figures 7-9. Similarly, **HEU** showed solid performances in terms of usability, with most results close to the optimal solutions.

Though **OPT** has better usability performance than **HEU**, it takes much longer time to get a solution for each testing case. As shown in Figures 10 and 11. It took at least 10 times longer time for **OPT** to find a solution than **HEU** did. For example, in Figure 10, when modality resource has a *loose* bound, **HEU** spent 0.13 seconds to find a solution with usability 3.6. **OPT** took 4.5 seconds to find an optimal solution with usability 3.9.

In all above experiments, CPU, memory and bandwidth requirements were using the same bound. In other words, in all above testings, they were all using *Tight* bound, *Average* bound or *Loose* bound. In the following, we check the performances for our solutions in general scenarios, in which some resource requirements are more tight or loose than others. For example, CPU is in the *Tight* bound while memory could be in the *Loose* bound. In Figures 12 and 13, we tested all the possible cases on the random network and the well-known network topology. We have the similar results regardless the network edge capacity range. Therefore, we only show the *Loose* edge capacity scenario on the random network and the *Tight* edge capacity scenario on the NSFNET. We observed that **HEU** performed well on both network topologies by delivering near-optimal solutions. Also **HEU** had better performance on the 50-node random network than on the 14-node NSFNET. The reason is that on large networks, **HEU** has more flexibility on finding feasible routings for the modalities using the shortest path algorithm.

Based on our comprehensive numerical evaluation, **OPT** and **HEU** both delivered good performances in terms of usability and running time. When network size and modality size is relatively small, **OPT** can provide an optimal solution. When the problem input size is large, **HEU** can provide efficient and effective near-optimal solutions in a short time, which is specially interesting and important for wireless network applications.

6. A CASE STUDY

In this case study, we consider a pervasive application, which supports a salesman to access information of products anytime and anywhere. This application includes a mobile device, e.g. a blackberry smart phone, which a salesman can take with at various customer locations. The mobile device supports four different input modalities: textual input is supported through a small keypad or a Bluetooth-enabled portable keyboard; speech-based input is enabled through a microphone; users can use a pen/stylus to perform pen-based interaction; and a Bluetooth mouse is supported to make a selection. The output modalities include speech, audio files, video files, graphical representations and textual information. In order to use above modalities, corresponding software must be installed. Each modality requires the installation of a driver, through which a hardware component is controlled. In addition, some modalities need some special software to input/output information, such as character recognition in the pen-based interaction, voice recognition in the speech-based input, an audio player for playing audio files and etc. The execution of these software components consumes valuable system resources. For example, the voice recognition software of *IBM Viavoice for Embedded devices* needs 700KB RAM. Furthermore, some modality has its QoS requirements. For example, if an audio file is transferred from a server located at the headquarter to the salesman's mobile device, the delay should not exceed certain seconds in order to provide a smooth playback. The resource demands and QoS requirements are systematically recorded in the resource profile and QoS profile, respectively, through an ontology language. As shown in Figure 14, the specification of the relations among modalities, interaction context, resources and QoS requirements is defined through CLMO.

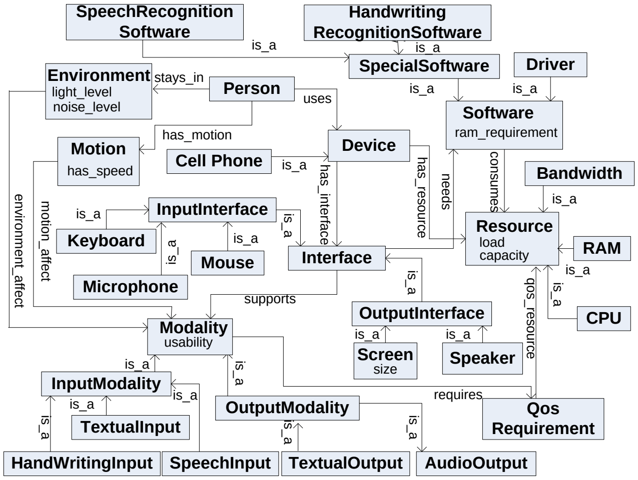

Figure 14. Part of the CLMO ontology definition for major classes and properties

Different interaction contexts can affect the usability of a modality. For example, based on CLMO, Figure 15 shows a scenario in which the salesman is walking. The following ontology shows the activity of *walking* reduces the usability of a hand writing input through pen/styles. On the other hand, speech-based input has a high usability value, equal to 1, when a user is walking. In addition, the adaptation should also consider resource and QoS profiles.

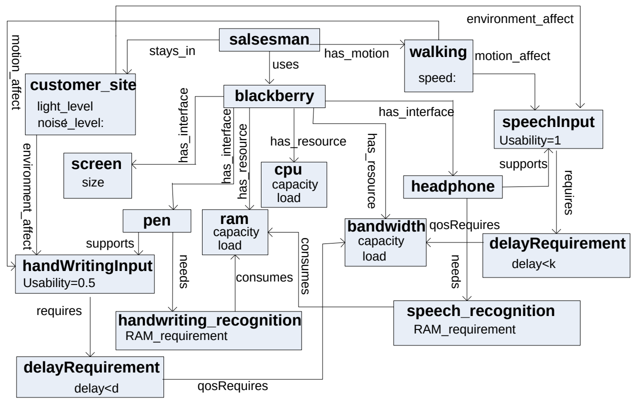

Figure 15. A working salesman scenario defined by CLMO

At the customer site, the salesman uses a wireless network connecting to the server. Emphasizing on a high mobility, he mainly uses pen-based or speech-based input. In case of a noisy environment, such as in a meeting, the salesman prefers the pen-based input since noise can significantly reduce the audio effect. Also, sensitive information should be input through pen, instead of speech, to avoid overhearing. The collected customer's requirements are sent to a database server. The feedback from the server, such as the availability of product, pricing options and etc, can be presented in the form of text, graphics or speech based on the availability of bandwidth and interaction context. For example, only the textual presentation can be offered with a limited bandwidth while a speech-based output is preferred when the salesman is in a quiet environment with busy hands. Video files may be associated with a product. The blackberry phone has the proprietary media player software installed to play video files. Since the video files can be very large in size, a 16 GB microSD card is attached. Audio quality/delay and synchronization of data are determined by the wireless connection. Depending on the bandwidth available, two adaption scenarios can be considered: 1) if the bandwidth is low, only texts and small-sized image files of the product description can be transferred, and 2) if the bandwidth is high, large-sized video files can also be transferred.

When a salesman is driving between customer and office sites, pen or keyboard cannot be used. He can use a Motorola Bluetooth enabled headset linked to the mobile phone to retrieve or access information through audio input/output.

At the office site, the salesman is directly connected to a high speed wired network. He can input data through a keyboard. Any sort of synchronization between the mobile device and centralized database would be extremely fast, assuming a T1 connection of 1.544 Mbps. Also delay in audio would be minimal. If there is a video chat conference between the salesman and the supervisor or other peer salesmen, it can be done with the best frame rate. Office space also has the flexibility to be equipped with a speaker that acts as an auditory output.

7. RELATED WORK

By providing a natural communication between the user and the computer, multimodal human-computer interaction has been growing fast. A majority of current approaches target to address issues in specific modalities [Dua06]. For example, Jaims and Sebe gave an extensive survey for multimodal human computer interaction from the perspective of computer vision [Jai05]. The techniques of facial expression recognition and vocal emotion recognition have been used in affective human-computer interaction [Pan03] to detect human emotion, which is an important feature in the interaction context. Instead of discussing the recognition of human emotions in a communications, this paper focuses on how to automatically select appropriate modalities based on human emotion and other features.

Several studies have been conducted to develop context-aware mobile applications [Sch99, Sie03], which are be able to adapt to dynamically changing environmental and physiological states. The above approaches focus on context recognition, instead of modality adaptation.

An important requirement for context-aware applications is the ability to adapt at run time. Several researchers [Cap03, Pol04] have worked on dynamically allocating system resources according to available resources and user preferences. Those approaches focus on reconfiguration of system software, instead of modalities.

Lemmela *et. al.* [Lem08] proposed a 5-step iterative process, which includes observing and analyzing interaction contexts, to design multimodal user interfaces. Based on the characteristics of different modalities, the designers can determine suitable modalities for different situations. Bouchet *et. al.* [Bou04] proposed a component-based approach for rapid development of multimodal interfaces. This approach includes two types of software components, i.e. elementary components for pure modalities and composite components for combining modalities. Those frameworks and models are valuable to design and develop multimodal interfaces, but they do not support automatic adaptation.

Rousseau *et. al.* [Rou05] developed a Multimodal Output Specification Tool, called *MOSTe*, which specifies multimodal output in terms of interaction components, interaction context and information units. A behavioral model based on election rules is used to define the adaptation upon different situations. Duarte and Carrico [Dua06] proposed a conceptual framework, i.e. FAME, for the development of an adaptive multimodal system. The FAME architecture uses different models to specify the features of a multimodal application from the perspectives of user, platform and environment. An innovative behavioral matrix is introduced to represent adaptation rules. Though rule-based specifications propose a simple reasoning with a low learning cost, they have the problems of the completeness and the coherence of the rules base [Rou05]. Distinct from the above approaches, this paper formalizes the adaptation as searching for an optimal set of modalities with the highest usability under a specific scenario. Furthermore, our work considers adaptation not only from the perspective of interaction context (i.e. user, platform and environment), but also from the aspects of system resources constraints and underlying QoS-assured routing.

8. CONCLUSION AND FUTURE WORK

Multimodal interfaces provide a natural communication between a computer and a user by supporting two or more communication modes. With the development of hardware and software techniques, such as voice recognition, many researchers focus on each mode independently and then fuse the results at the application level. However, it stills leaves as an open problem of addressing the modality adaptation in the process of multimodal interface design. Recently, several frameworks and models have been proposed to support adaptation through rule-based specifications. Rule-based adaptation may not be complete to different scenarios and different rules could conflict with each other. This paper presents a novel approach based on integer linear programming. Our approach considers an adaptation from three perspectives, i.e. interaction context, system resources and QoS requirements, which crosses the application, system and network layers. Different from previous work, this paper formalizes the adaptation issue as searching for a set of input/output modalities that produce the highest usability with satisfied resource and QoS requirements. Given a specific interaction scenario, both an optimal solution and a heuristic algorithm are developed to automatically select an appropriate set of modalities combinations. The numerical evaluation shows good performance of the proposed solutions.

This paper focuses on selecting most appropriate modalities, which fit an interaction context with QoS and system resource requirements observed. After an appropriate modality is selected, a multimodal interface needs to further determine the information presentation, such as what information to present (i.e. only output relevant information), how to present it (i.e. color and font size), when to present it (i.e. synchronization with other information). In the future, we will consider combining modality adaptation with content adaptation.

9. ACKNOWLEDGMENTS

This work is supported in part by ND EPSCoR State Seed Grants #FAR0015845 and #FAR0015846.

10. REFERENCES

[Abo98] Abowd, G. D., Dey, A. K., Orr, R. Brotherton, J.,Context-awareness in wearable and ubiquitous computing, Virtual Reality,3,(3), Pages 200 – 211, 1998

[And04a] R. Anderson, C. Hoyer, C. Prince, J. Su, F. Videon, and S. Wolfman, "Speech, Ink and Slides: The Interaction of Content Channels", *ACM Multimedia*, 2004.

[And04b] R. J. Anderson, C. Hoyer, S. A. Wolfman, and R. Anderson, "A Study of Digital Ink in Lecture Presentation", *Proc. CHI'04*, pp. 567-574, 2004.

[Bou04] J. Bouchet, L. Nigay, and T. Ganille, "Icare Software Components for Rapidly Developing Multimodal Interfaces", *Proc. ICMI'04*, pp. 251-258, 2004.

[Cap03] L. Capra, W. Emmerich and C. Mascolo, "CARISMA: Context-Aware Reflective mIddleware Systems for Mobile Applications", *IEEE Transactions on Software Engineering*, Vol. 29(10), pp.929-945, 2003.

[Dra04] P. Dragicevic and J. D. Fekete, "The Input Configurator Toolkit: Towards High Input Adaptability in Interactive Applications", *Proc. AVI'04*, pp.244-247, 2004.

[Dua06] C. Duarte and L. Carrico, "A Conceptual Framework for Developing Adaptive Multimodal Applications", *Proc. IUI'06*, pp.132-139, 2006.

[Fli03] F. Flippo, A. Krebs, and I. Marsic, "A Framework for Rapid Development of Multimodal Interfaces", *Proc. ICMI'03*, pp. 109-116, 2003.

[Gar90] M. Garey and D. S. Johnson, "Computers and Intractability: A Guide to the Theory of NP-Completeness", *W. H. Freeman and Co. New York*, NY, 1990.

[Gru95] T. R. Gruber, "Principles for the Design of Ontologies Used for Knowledge Sharing," *International Journal Human-Computer Studies*, 43(5-6), pp.907-928, 1995.

[Hec05] D. Heckmann, B. Brandherm, M. Schmitz, T. Schwartz, and B. M. von Wilamowitz-Moellendorf, "GUMO - the general user model ontology", *Proc. the 10th International Conference on User Modeling*, pp. 428–432, 2005.

[Jai05] A. Jaimes and N. Sebe, "Multimodal Human Computer Interaction: A Survey", *Proc. IEEE International Workshop on Human Computer Interaction*, in conjunction with ICCV' 20, 2005.

[Kai07] E. Kaiser, P. Barthelmess, C. Erdmann, and P. Cohen, "Multimodal Redundancy Across Handwriting and Speech During Computer Mediated Human-Human Interactions", *Proc. CHI'07*, pp.1009-1018, 2007.

[Kor01] T. Korkmaz and M. Krunz, "A Randomized Algorithm for Finding a Path Subject to Multiple QoS Requirements", *Computer Networks*, Vol. 36, pp. 251-268, 2001.

[Kor06] N. Korfiatis and I. Constantiou, "Socially Targeted Mobile Services: Towards An Upper Level Ontology Of Social Roles For Mobile Environments", *Proc. the 14th European Conference of Information Systems*, 2006.

[Lem08] S. Lemmela, A. Vetek, K. Makela, D. Trendafilov, "Designing and Evaluating Multimodal Interaction for Mobile Contexts", *Proc. ICMI'08*, pp. 265-272, 2008.

[Obr07] Z. Obrenovic, J. Abascal, and D. Starcevic, "Universal Accessibility as A multimodal Design Issue", *Communication of the ACM*, Vol.50(5), pp.83-88, 2007.

[Ovi97] S. L. Oviatt, A. DeAngeli, and K. Kuhn, "Integration and Synchronization of Input Modes During Multimodal Human-Computer Interaction", *Proc. CHI'97*, pp.415-422, 1997.

[Ovi99] S. Oviatt, "Ten Myths of Multimodal Interaction", *Communications of the ACM*, Vol. 42(11), pp. 74-81, 1999.

[Ovi02] S. Oviatt, "Breaking the Robustness Barrier: Recent Progress on the Design of Robust Multimodal Systems", *Advances in Computers*, Vol.56, pp.305-341, 2002.

[Ovi04] S. Oviatt, T. Darrell, and M. Flickner, "Multimodal Interfaces that Flex, Adapt, and Persist", *Communication of the ACM*, Vol.47(1), pp.30-33, 2004.

[OWL] OWL Web Ontology Language: http://www.w3.org/TR/owl-features/

[Pan03] M. Pantic, L.J.M. Rothkrantz, "Toward an Affect-Sensitive Multimodal Human-Computer Interaction", *Proceedings of the IEEE*, 91(9), pp.1370-1390, 2003.

[Pol04] V. Poladian, J. P. Sousa, D. Garlan, and M. Shaw, "Dynamic Configuration of Resource-Aware Services", *Proc. the 26th International Conference on Software Engineering*, 2004.

[Pre04] D. Preuveneers, J. Van den Bergh, D. Wagelaar, A. Georges, P. Rigole, T. Clerckx, Y. Berbers, K. Coninx, V. Jonckers and K. De Bosschere, "Towards an extensible context ontology for ambient intelligence", *Ambient Intelligence*, pp.148-159, 2004.

[RDF] Resource Description Framework (RDF): http://www.w3.org/RDF/

[Rou05] C. Rousseau, Y. Bellik, and F. Vernier, “Multimodal Output Specification / Simulation Platform”, *Proc. ICMI’05*, pp.84-91, 2005.

[Sch94] B. N. Schilit, N. I. Adams, R. Want, “Context-Aware Computing Applications”, *Proc. Of the Workshop on Mobile Computing Systems and applications*, 1994.

[Sch99] A. Schmidt, K. A. Aidoo, A. Takaluoma, U. Tuomela, K. V. Laerhoven, W. V. de Velde, “Advanced Interaction in Context”, *Proc. Handheld and Ubiquitous Computing*, pp.89-101, 1999.

[Sie03] D. Siewiorek, A. Smailagic, J. Furukawa, A. Krause, N. Moraveji, K. Reiger, and J. Shaffer, “SenSay: A Context-Aware Mobile Phone”, *Proc. 7th IEEE International Symposium on Wearable Computers*, 2003.

[Str03] T Strang, C Linnhoff-Popien, K Frank, "CoOL: A context ontology language to enable contextual interoperability", *Distributed Applications and Interoperable Systems*, pp.236-247, 2003

[Wic99] C. Wickens and J. Hollands, *Engineering Psychology and Human Performance*, Prentice Hall, 1999.

[Xue07] G. L. Xue, A. Sen, W. Y. Zhang, J. Tang, and K. Thulasiraman, “Finding a Path Subject to Many Additive QoS Constraints”, *IEEE/ACM Transactions on Networking*, Vol. 15, pp.201-211, 2007.