Profile-based Data Diffusion in Mobile Environments

Thomas Buchholz, Iris Hochstatter, Georg Treu
Mobile & Distributed Systems Group, Institute for Informatics
Ludwig-Maximilian-University Munich
Oettingenstr. 67, 80538 Munich, Germany
Email: {thomas.buchholz/iris.hochstatter/georg.treu}@ifi.lmu.de

Abstract—People on the move often are interested in information about their close proximity. This information can either be retrieved from static nodes or other mobile nodes spreading information via wireless technologies. This approach of data diffusion potentially overloads the mobile device if no filtering is applied. Filtering mechanisms, however, cannot rely on the existence of a common information model among the peers.

In this paper, we present an approach for profile-based filtering which does not require shared knowledge among the nodes. Instead, our system can translate between competing taxonomies. It uses synonyms and a tolerant search algorithm to cope with morphological and lexical variations and structural differences of the taxonomies. A prototype of our approach was implemented and successfully tested.

I. INTRODUCTION

People on the move often are interested in information about their close proximity. Data diffusion has been proposed to reach the local spreading of information [1], [2]. The main principle is that static server nodes announce information via local wireless radio technologies like Bluetooth or IEEE 802.11. Mobile nodes listen to this information and cooperate to disseminate the information by word of mouth. This approach of data diffusion potentially overloads the mobile device if no filtering is applied. Filtering mechanisms, however, cannot rely on the existence of a common information model among the peers. Therefore, we employ a tolerant matching mechanism by using synonyms and a tolerant Breadth-First-Search to account for differences in the structure of the competing information models.

II. APPROACH FOR TOLERANT PROFILE-BASED DATA EXCHANGE

Our protocol exchanges data items between two nodes when they come into radio range. These can either be two mobile nodes or an immobile and a mobile node. A data item consists of its payload, i.e. the data item content, and a description or classification of the payload, i.e. the data description. Whenever two nodes meet, the following protocol is carried out: (1.) Profile exchange: Profiles are transmitted between the peers. (2.) Matching: Both handhelds search in their cache if data items are available that match to the profile of the peer. (3.) Data exchange: Matching data items, if any, are transferred to the peer.

As the set of possibly interesting data items and their types is unlimited, we do not assume that there will be one global information model describing all possible data items. Nevertheless, the mobile devices have to be able to decide if the data description of a data item matches to a profile. Therefore, we need some kind of shared understanding about previously unknown data descriptions.

The representation of knowledge and automatic extraction of new knowledge has been solved in many different ways, e.g. taxonomies, thesauri, topicMaps, semantic networks, entity-relationship models, ontologies etc. [3]. In our prototype, we used taxonomies. They are capable of expressing hypernyms and hyponyms. The hypernym relation expresses that a class A is a more general form of class B. The inverse of a hypernym is called a hyponym or specialization. Figure 1 shows an example taxonomy. A taxonomy can be understood as a tree of concepts. Concepts that are above others are more general than the lower concepts. Taxonomies do not allow for automatic reasoning. However, they are well suited for the classification of data items and their representation is understandable by humans.

![Example taxonomy](image)

As an entry in a profile can simply be described by a sequence of references to successive nodes in a taxonomy starting from the top. One example is Shopping.Apparel OUTERWEAR.Sweaters. A profile is a set of such entries. Since there may exist multiple competing taxonomies, we employ a tolerant matching algorithm using schema matching. Schema matching applies heuristics to find similarities between different schemata [4] and is often used in Information Retrieval to extract useful information from data warehouses. In contrast to ontology-based matching, different resources do...
not have to share the same ontology. The shared knowledge between resources is not explicitly stated with the help of an ontology language but is reduced to a more informal source of common ground, such as e.g. the semantic relationship of words in natural languages. In our system, each mobile device maintains a local taxonomy and tries to map incoming data descriptions from a probably different schema.

Matching can be difficult due to morphological, lexical and semantical variations. Examples for morphological variations are e.g. the plural or possessive forms of nouns. In our example taxonomy in figure 1, measures must be taken that also the keyword sweater is recognized though the taxonomy contains the word sweaters. Lexical variations mean that a lot of synonyms exist for a word. For example, mobile nodes might ask for data items about "clothes" while the example taxonomy uses the term "apparel". Semantical variations refer to the fact that some words have more than one meaning according to the context. Apparently, the noun "bank" stands for two different concepts in the sentences "Frank sits down on a bank" and "Mary brings her money to the bank". Semantical variations are hard to cope with in an automatic way. All synonyms and morphological variations of the keywords used are retrieved and added to the taxonomy before it is brought onto the handheld. Whenever the danger arises that semantically varied synonyms are considered, the user is interactively prompted to decide whether he wants the synonym to be added.

Another difficulty during the matching process on the handheld might be that two mobile nodes use different structures for their taxonomies. For example, the exact same branch apparel might be a child node to root on one handheld and not to shopping like on another handheld. We do not only compare keywords on equal tree levels, but use a Breadth-First Search (BFS) on the taxonomy: starting from root, the algorithm looks for matches on keywords on the same depth of the tree before testing the children. Thus, any possible ambiguities are resolved because higher nodes are given higher priority in the search. If unsuccessful, the BFS interrupts after a given amount of tree levels and skips subpaths of the reference. In this algorithm, generalizations or specializations of a given keyword are recognized, as they are parent respectively child nodes and we can thus identify the relevance of the incoming keyword to the own taxonomy.

After having successfully mapped a profile entry to the taxonomy, the next step is to determine the relevance of cached data items for a given set of profile entries. To calculate this degree of approximation, we use the following algorithm: Let us assume that the profile contains the data description $d_p$ and the cached data item with the description $d_c$ is to be mapped onto this. The path from the root to $d_p$ within the taxonomy is denoted with $p_p$, while the path to $d_c$ is named $p_c$. The respective lengths of these paths are $l_p$ and $l_c$.

The following formula calculates the relevance of $d_c$ to $d_p$, based on a given taxonomy:

$$m(d_c, d_p) = \begin{cases} l_p / l_c & d_c \in p_p \lor d_p \in p_c \\ 0 & d_c \notin p_p \land d_p \notin p_c \end{cases} \quad (1)$$

An optimal match (equality) is reached at a value of 1.0. A value greater 0 and less 1.0 signals a specialization. This means that the description of the cached data item is more specific than the data description that is listed in the profile. If the cached data item is more general than the data specified in the profile of the peer, the value lies above 1.0. A value of 0 signals that the two entities can not be compared. A data item is considered relevant for a given device profile, if at least one of the data descriptions forming the profile matches the description of the cached data item for a given upper and lower bound. For example, a device profile might contain the data descriptions Transportation.Public and Shopping.Apparel.Outerwear.Sweaters.

The device profile is transmitted together with the lower matching bound 0.5 and the upper matching bound 2. The device that receives this device profile has data items with the following descriptions in its cache: Transportation.Public.Bus.Schedule, Shopping.Apparel.Outerwear, and Shopping.Apparel.Underwear. The data item with the description Transportation.Public.Bus.Schedule is exchanged because it matches to Transportation.Public with $\frac{3}{4}$. Shopping.Apparel.Outerwear is transmitted since it fits to Shopping.Apparel.Outerwear.Sweaters with $\frac{1}{4}$. Shopping.Apparel.Underwear does not lie on the path of any data description in the profile and is not exchanged.

III. IMPLEMENTATION

Our concepts were evaluated by implementing a prototype in Java as a middleware between applications and the network connection. The middleware was tested on Compaq iPAQs H3970 running a linux operating system. To access the Bluetooth interface the JSR-82 API for Bluetooth [5] was used.

The heart of our profile-based exchange of information is the taxonomy since it forms the basis for specifying the profile and classifying data items.

To enable a handheld to map incoming data descriptions specified in an unknown taxonomy to the own taxonomy, we use the synonym database WordNet [6] which offers a complete reference for synonyms in the English language. Unfortunately, WordNet is much too large to be installed on a mobile device. However, we do not need all synonyms of the English language. The only interesting part are the synonyms of the keywords that are used in the profile. Therefore, we decided to preprocess the user’s profile like depicted in figure 2.

The profile is expressed in an XML file that is fed into the Taxonomy Creation System (TCS). The TCS parses all entries in the XML file and retrieves synonyms for all keywords in the profile from the WordNet synonym database. After having processed all entries in the profile, a complete taxonomy with all synonyms exists. This is stored on the user’s handheld in a component called taxonomy dictionary.

Besides the taxonomy dictionary, four other important components exist on the handheld: the taxonomy handler, the profile, a cache, and a broker. Since the taxonomy dictionary is only a passive container for the taxonomy and the respective
synonyms, another active component is needed, the taxonomy handler. It provides the functionality to translate incoming data descriptions with the help of the synonyms to the local taxonomy and calculates the relation between data descriptions. The profile is the container for the interests of the user and the cache stores collected data items. The broker is in charge of communicating with other mobile nodes and coordinating the exchange process.

Figure 3 depicts how the components interact to exchange data items with another device. Whenever two devices meet, the two brokers on them establish a connection and exchange the interest profiles with upper and lower bounds for the matching algorithm. The internal process is explained only for one device, device B, since on device A exactly the same workflow is carried out. After having received the profile and the matching bounds, the profile is given to the taxonomy handler. The taxonomy handler translates the descriptions of interests found in profile A into descriptions corresponding to the own taxonomy with the help of the synonyms in the taxonomy dictionary. Afterwards, the taxonomy handler passes a set of data descriptions corresponding to the own taxonomy to the broker. The broker queries the cache for data items that match the descriptions according to the matching algorithm in section II. To fulfill its task the cache makes use of the taxonomy stored in the taxonomy dictionary component via invocations of methods provided by the taxonomy handler component. Most importantly, the relevance of incoming data descriptions is figured out. Then the cache responds with all matching data items. These data items are given from broker B to broker A on the other device. Broker A answers with data items that might be interesting for device B. Since these data items are classified with descriptions corresponding to the taxonomy of device A, the descriptions need to be translated to integrate the data items into the local taxonomy. After the translation the data is stored in the cache.

IV. RELATED WORK

Data diffusion has initially been employed in sensor networks, e.g. [7]. Systems that are similar to ours are 7DS [1] and Usenet-on-the-fly [2]. In contrast to our system, [1] and [2] assume a common shared information model among all participating nodes. An alternative for mobile nodes to find information is through resource discovery. Three main approaches can be distinguished: central repositories, flooding of requests, and building overlays. See [8] for a discussion of these approaches. All of these approaches are either infeasible or inefficient in our setting and are therefore inappropriate.

V. CONCLUSION

In this paper, we have described how information overload of mobile devices can be avoided in data diffusion systems through profile-based filtering. Though profiles are based on taxonomies, no shared knowledge between the mobile nodes is assumed. Instead, our system is able to tolerantly translate among multiple competing taxonomies. This is achieved by considering synonyms of keywords and by using an extended Breadth-First-Search to account for structural differences between the taxonomies.

REFERENCES