Exploiting Social Feedbacks in Urban Environments with SAPERE

B. Anzengruber*, G. Castelli†, A. Rosi†, T. Schmittner*, A. Ferscha*, F. Zambonelli†
* Institute for Pervasive Computing, Johannes Kepler University Linz, Austria
Email: surname@pervasive.jku.at
† Dipartimento di Scienze e Metodi dell'Ingegneria, Universitá di Modena e Reggio Emilia, Italy
Email: name.surname@unimore.it

***Abstract*—People living in modern cities are exposed to rapidly spreading display technologies (e.g., smartphones, public screens, etc.). We envision these displays to be context-aware and self-coordinating and as a consequence capable of implicitly interacting with other system components. Therefore people can be served with individually customized content and consume smart services tailored to their needs. In order to implement such systems, we rely on SAPERE, that provides means of spontaneous, local and distributed interactions between spatially or semantically collocated devices and their respective users. To demonstrate the potential of the SAPERE framework, we have developed a social feedback application enabling user-specific content supply, and influencing the behavior of system users by providing (social) feedback on their current actions.**

***Index Terms*—pervasive middleware; spontaneous interaction; smart environment; social feedback; SAPERE; ecosystem.**

I. Introduction

Display systems have become an omnipresent "pervasive technology" that penetrates our daily lives [1], [2], [3]. It is a matter of fact that an increasing number of our everyday activities involve interacting with private displays or even large public display systems deployed in our immediate environment. Most commonly, such interactions consist of the one-sided conveyance of information and other content, however explicit, gesture-based interactions between user and display are becoming increasingly prominent.

We envision display systems of a heterogeneous- and distributed nature to be context-aware, self-coordinating, able to sense their location and immediate surroundings and therefore to be capable of implicitly interacting with present users and passersby. As a consequence these "display ecosystems" consisting of numerous distributed, though networked and self-aware displays, are considered to serve people with individually customized content and provide them with useful and smart services [4], [5].

In particular, in this paper we present a social feedback application where public displays guide users to individually optimized food providers (e. g., restaurants, pub, cafeteria, etc.). This selection is based on (i) the food preferences of users and on (ii) the estimated waiting time at the various lunch locations. In addition, the proposed application is influenced by the idea of "social capital" [6] and considers "social feedback" – which is provided by the society (i.e., other users of the system) – to be an influential factor for the individual behavior of humans.

Developing such a social feedback application involves dealing with the dynamism of people moving in an environment, or in distributed sites. Challenges in such a setting include large number of devices handshaking and exchanging information as well as device heterogeneity in terms of hardware and software components. From the above requirements we opt for implementing this application with SAPERE [7] – an approach modeling and deploying components as autonomous actors in an ecosystem that is populated by services, data sources, and pervasive devices. The decentralized nature of the SAPERE paradigm allows massive scale without "single-points-of-failures" and exploitation of local interactions between distributed components leads to the emergence of self-* properties that can be used to maintain functionality and connectedness of the system.

By instantiating public and privately owned displays as SAPERE nodes, the proposed application is able to provide tailored content and dynamically adapted recommendations based on the preferences of users currently in front of public displays. Due to the self-aware capabilities of the system components users can be dynamically steered to arbitrary locations since each and every node in the system knows both its absolute position and its relative position with regard to all the neighboring displays.

The remainder of this paper is organized as follows: Section II provides an overview of SAPERE. Section III presents the middleware implementing the SAPERE concepts. Section IV introduces and models a social feedback application, while Section V provides an implementation in SAPERE terms. In Section VI the related work is discussed. Finally Section VII concludes this paper.

II. SAPERE in a Nutshell

This section provides a concise overview of SAPERE describing its general architecture and introducing a reference operational model.

A. The SAPERE Architecture

SAPERE considers structuring a pervasive service environment as a non-layered spatial substrate, mapped above the actual network infrastructure.

For any individual (i.e., devices, users, software services) in the environment SAPERE adopts a common modeling and treatment: each of them has an associated self-descriptive

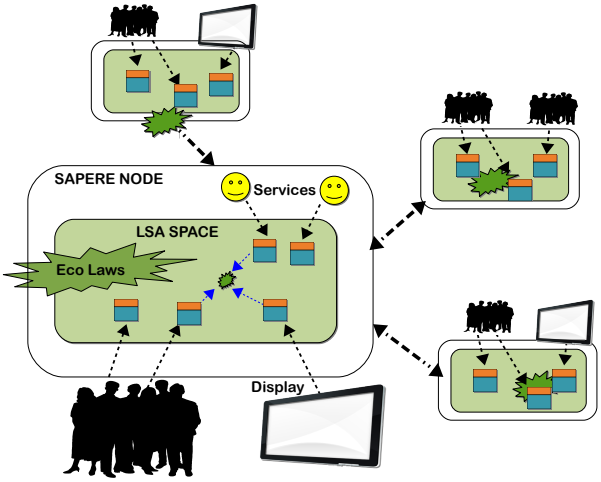


Fig. 1. SAPERE Operational Model

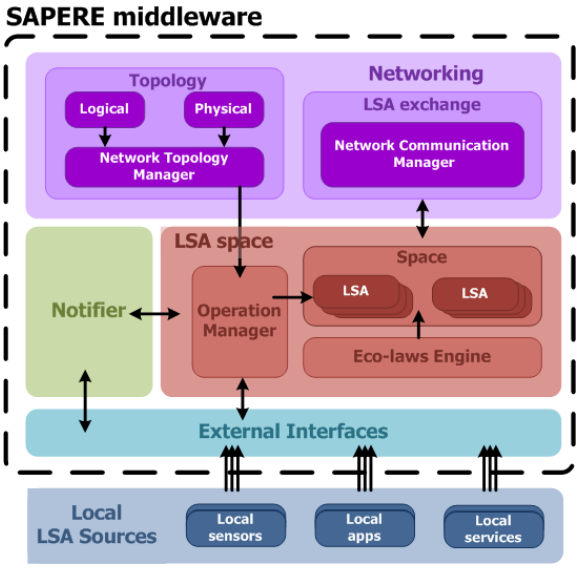


Fig. 2. The middleware Architecture

semantic representation, called *LSA* (Live Semantics Annotation). To account for the high dynamics of the components represented, such annotations are defined as living, active entities, tightly associated to the component they describe, and capable of reflecting their current situation and context. Namely, the interaction of any individual with the ecology is enacted through their LSA, by modification/observation of their structure and of the information they hold.

The substrate embeds the basic laws of nature or "*eco-laws*" which rule the activities of the system. Eco-laws define the basic policies to govern sorts of virtual chemical reactions among LSAs, thus enforcing dynamic concept-based (i. e., semantic and goal-oriented) networking, composition, and coordination of data and services in the ecosystem to establish bonds – or new relations – between entities, produce new LSAs – or information – and diffuse LSAs in the network.

Adaptivity in SAPERE is not only restricted to the capabilities of an individual component, but rather arises from the overall dynamics of the ecosystem. In particular, adaptivity is ensured by the fact that any change in the system (as well as any change in its components, as reflected by dynamic changes in their LSAs) will reflect in the firing of new chemical reactions, thus possibly leading to the establishment of new bonds and/or in the breaking of some existing bonds between components.

B. The SAPERE Operational Model

The SAPERE environment considers a dynamic set of computational nodes mapped above the actual pervasive network infrastructure, which is composed by a possibly very dense set of hardware devices (see Figure 1).

In SAPERE any device provided with sufficient computational power to handle an LSA-space is eligible to constitute a node of the network enabling personal computers, notebooks, modern tablets and smartphones to be part of the SAPERE environment.

From the operational point of view, all SAPERE nodes are perfectly replaceable since they can provide the same services and set of functionalities. Despite such formal equality in purposes, nodes might implement different behaviors in order to account for particular service requirements or specific hardware limitations (e. g., a node on a mobile device could assume a behavior accounting for constrained power supply).

Each SAPERE node embeds an LSA-space, in which self-adaptive coordination mechanisms take place so as to mediate the interactions between components. Whenever a component approaches a node, its own LSA is automatically injected into the LSA-space of that node, making the component part of that space and of its local coordination dynamics and gets immersed in the ecosystem. Similarly, when a component moves away from a node, its LSA is automatically removed from that space.

III. The SAPERE Middleware

The SAPERE middleware is realized as a minimal middleware spread over a network of nodes, each embedding an active tuple space in which LSAs are reified in the form of tuples and the eco-laws as rules happening in there. This software framework is written in Java, and an Android version is available as well.

Figure 2 depicts the middleware architecture. Each node is made up of four components:

- The *external interfaces* which provide APIs for agents to generate and manage LSAs
- The *LSA-space* which represents an active tuple space
- The *Notifier* that manages events happening to LSAs and notifies the agents
- The *Networking* that builds the network topology and rules the LSAs exchange with other nodes

The **LSA-space** is realized as a lightweight tuple space that stores local LSAs and executes eco-laws over them. The core component is the *Space*, a passive component that stores LSAs and provides a primitive access to them. The *Operation*

Manager and the *Eco-laws Engine* get mutual access to the Space to submit operations and execute eco-laws respectively. The Operation Manager submits one of the following four supported operations to the Space: *inject*, *remove*, *read* and *update*. While the Space effectively executes the operations to the LSA collection, the Operation Manager, for each operation submitted, takes care of interaction with the *Notifier* and of properly forwarding operations to the Space. The *Eco-laws Engine* gets periodically activated in order to execute eco-laws on the collection of locally stored LSAs.

The **Notifier** components take care of managing events happening to LSAs (e. g., bonds, de-bonds, etc.) and notifying the agents in charge of managing the corresponding LSAs. To realize the *event notifier pattern* we defined: *(i) events* to be announced, *(ii) subscriptions* to record the interest of a subscriber to a particular event and *(iii) filters* responsible for discarding events not of interest to a subscriber. Events are fired by the Space object as operations on LSAs are performed, for each fired event the Notifier checks if there are subscribed listeners for that event and invokes the filters to detect the specific subscriber that is to be notified.

The **Networking** module manages interactions with other SAPERE nodes that is building and maintaining a topology of the SAPERE network enabling the exchange of LSAs between neighbors. The process of managing the topology of the network is distributed on each local node, where the local *Network Topology Manager*, depending on the node configuration, instantiates two independent processes: the *Logical Topology Manager* and the *Physical Topology Manager*. The Physical Topology Manager is in charge of detecting neighbor nodes that are in physical proximity (the actual implementation lays over Bluetooth technology).

From another side, the Logical Topology Manager is in charge of exploring topologies that are not directly related to the spatial proximity, such as social proximity or hierarchical node organizations. At this regard, in [8] we further elaborate on innovative topology strategies based on the exploitation of social networks for shaping relationships between network entities. In order to implement applications, services and libraries relying on the current topology structure, an LSA containing neighbor details is injected for each neighbor in the conciliated table in the local LSA-space.

IV. A Social Feedback App

In this section we describe a pervasive application based on social feedback motivating how user behavior can be influenced by introducing a reward system based on *"Social Capital"* and introduce the key ingredients to model the app exploiting the aforementioned concepts.

A. Modeling a Social Feedback App

"Social feedback" describes "society's" response to a specific behavior of users, where society is made up of other users in a defined neighborhood. Social feedback is either positive if the user's behavior benefits the overall system goals or negative otherwise.

To this end, we take the notion of social feedback and instantiate it to model a social feedback application in an *ecosystem of displays*. This means, that a user "earns" social capital based on his behavior and contextual information – such as the current system state or the behavior of other, related ecosystem nodes. We continue to describe the necessary components of such an application and explain, how they may be implemented.

The node's **behavior** describes the way in which this ecosystem components interacts with the system and its neighbors. In order to evaluate the current behavior and to allow computation of its value to society, a system-specific set of possible behaviors $b_0 \dots b_n$ that a node in the system may adopt, is necessary. Behaviors may be very application-specific (e. g., for an urban car rental application b_x = "Return rental car to registered docking station"), system specific (e. g., b_x = "broadcast to all physical neighbors within communication range") or compound (i. e., a number of sub-behaviors that are evaluated as a group). However, any such behaviors are represented internally as a weighted function of the effort to adopt a specific behavior, the personal response to the behavior as well as society's response to it: $b(a) = (w_{Effort}(a) * Effort) + (w_{SocR}(a) * SocR) + (w_{PerR}(a) * PerR)$, where "$a$" is a user and "$b(a)$" an associated behavior.

The **utility function** describes the value of a given behavior to the system. It takes a user's current behavior as input and computes the amount of social capital gained or lost as a result of the given behavior. The ecosystem and the neighboring nodes of the current user compute this utility separately since the goals of the system may differ from those of an individual node. This behavior models real world societies, in which individuals do not always adopt behaviors that benefits society as a whole, but rather themselves.

The **individual social capital** or "SCI" is individually maintained by all ecosystem nodes and describes the current amount of the node's social capital.

In addition to the SCI, we also track the **overall social capital** or "SCO" of all nodes in the system. This is done in order to put the SCI values into a system–wide perspective. Based on the knowledge of how much capital is currently distributed in the system, mechanisms to adjust for phenomena like inflation or speculation (i. e., attempting to exploit implementation-specific weaknesses) can be implemented to keep the system valid (similar as with real-world currencies).

The **conversion function** allows transference of social capital into a tangible motivation for the user to utilize the system. The most obvious example of such motivation is a monetary remuneration, however other motivations such as application, and/or user specific bonuses, are possible (e. g., gift vouchers, parking time, reduced prices, regular rewards, etc.). The conversion function is dynamic and may change depending on the current system state (i. e., context) in order to keep the system valid.

B. App description

Developed for city-scale environments, we exploit the massive dissemination of display technologies to steer users towards an individually optimized food provider (e. g., restaurants, take-away, bars, etc.) that is known to the system. This optimization is computed based on *(i)* a user's food preferences (e. g., vegetarian, sushi, sandwiches. . .) and on *(ii)* the estimated waiting time at the various lunch locations – abstracted from the current number of customers and knowledge of past waiting time distributions. A location recommendation is computed with the general goal of minimizing the total waiting time of all users in the system (thus benefiting society). This recommendation is offered to the user together with steering information from the current position to the food provider. If the user acts as recommended by the system and adopts the associated behavior of eating at the optimal location, he is rewarded in the form of an increase of his SCI. However, if the user adopts another behavior (i. e., visits a lunch location other than the one recommended), his social capital will be reduced. High social capital can finally be converted to price reductions motivating users to utilize the system correctly.

V. Implementing a Social Feedback App in SAPERE

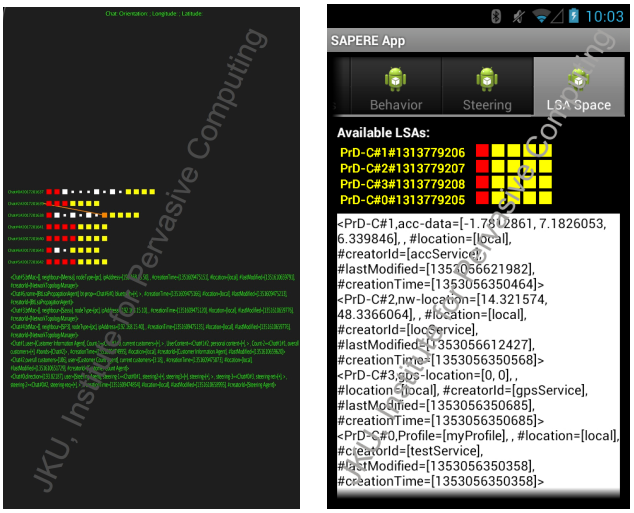
In this section we realize the already introduced social feedback App in SAPERE terms, provide some visuals from the prototype and describe our experiences from a developer's viewpoint.

A. Developing the App in SAPERE

Using the introduced SAPERE structures and software modules, we instantiated an ecosystem of displays testbed and developed an application prototype based on the presented social capital model. The SAPERE ecosystem of displays testbed consists of *(i)* multiple large scale public display devices mounted on pedestals that have been equipped with sensor technologies to detect each display's contextual state and *(ii)* private (i. e., user-owned) display devices such as smartphones, both of which run the SAPERE app. The testbed's public displays are presence-aware and able to detect private displays in their environment. In the developed prototype, we deploy 5 dedicated SAPERE agents to handle interactions and computations on public and private displays:

The **CustomerAgent** (running on both public and private displays) handles the exchange of user profiles once a private display moves into communication range of a public display. On the public display, this agent injects the received user profile LSA into the local LSA-space. On a private display, it updates the user profile to account for changes of the user's social capital.

The **CustomerInformationAgent** extracts user–specific information from the injected user profile LSA and passes this data to the CanteenAgent. It listens for steering requests and on reception creates a respective steering LSA. If the local display is a lunch location display, the CustomerInformationAgent further listens for the updated social capital of a user and propagates this data to the private display's CustomerAgent.



(a) (b)

Fig. 3. The SAPERE space view displays spatial properties and SAPERE–related constructs (a) on a public display (b) on a private one (a mobile phone).

The **CanteenAgent** receives user-specific data from the CustomerInformationAgent and initiates searches for optimal lunch locations – given the user's food preferences extracted from the LSA. The agent computes the current monetary value of the user's social capital based on the utility and conversion functions. Once a certain amount of time has expired, the CanteenAgent selects the optimal lunch location from the received responses and returns all of the computed information to the CustomerInformationAgent. If the local display is a lunch location display, the CanteenAgent computes the updated "SCI" of the user and the new "SCO" based on the current utility and sends this information to the customer's private display (where the user profile is updated accordingly) as well as to the visualization. Also, the CanteenAgent at a lunch location listens to LSAs containing a lunch location search. If such a request is received, the local lunch location's matching to the food constraints included in the LSA is computed and returned to the requesting node.

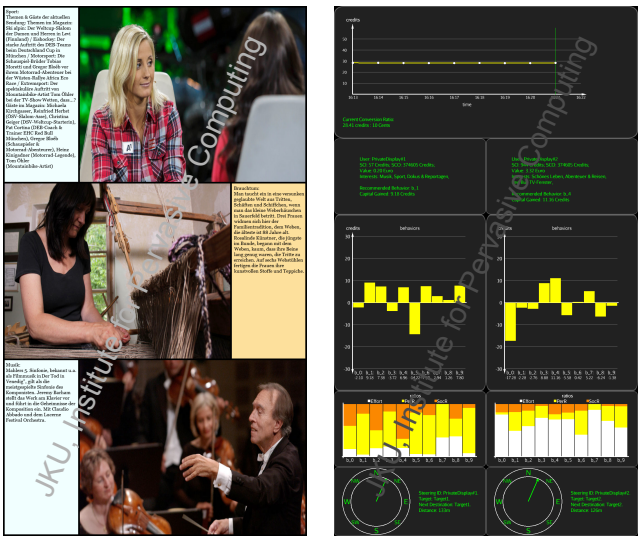
The **SteeringAgent** propagates steering requests through the ecosystem, waits for a response and forwards it to the CustomerInformationAgent.

The **CustomerCountAgent** is only active on a lunch location's public display. It updates an LSA encoding the current customer count at this lunch location. Once the private display is again sensed by the lunch location's public display (i. e., once the user leaves the canteen) the current customer count for this location is decremented.

B. App Prototype

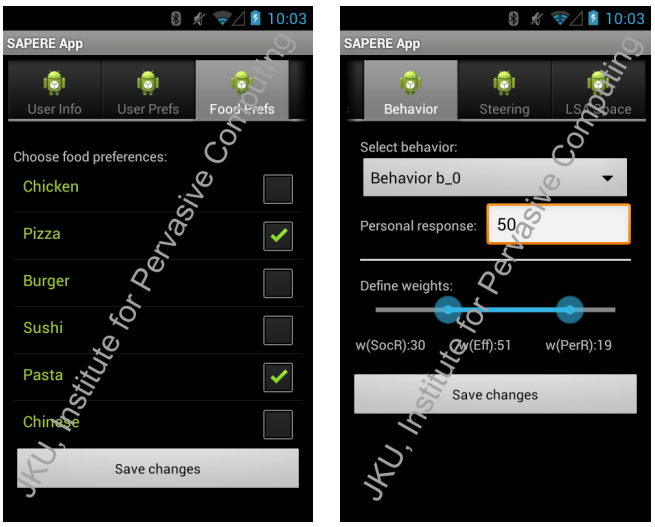
The SAPERE social feedback application running on top of our display ecosystem testbed offers three different views (see Figure 4), each tailored to a different contextual state:

SAPERE space view – Figure 3(a) and 3(b) In this operation mode, the application displays the LSA-space of the local SAPERE node both graphically and textually. The graphical



(a) (b)

Fig. 4. (a) The Content View: contains advertisements, user–matched content and other context–dependent information. (b) The Social Capital View: displays the conversion function, behavior ratios, behavior recommendations and steering information for two users.



(a) (b)

Fig. 5. The application views on a private display: (a) user food preferences and (b) general application settings.

representation of the space models LSAs as sequence of color-coded quads, whereby quads represent an LSA property the type of which is indicated by color. Bonds are represented as associated lines connecting the bonded LSAs. This view also displays the local SAPERE name and the spatial context of this node (orientation, longitude, latitude).

Content View – Figure 4(a) This operation mode represents the application's main view. If no other system component is in its interaction range, it acts as a public signage system and displays content such as advertisements, local news or weather. If an interaction partner is present, the application extracts this user's interests from the transmitted user LSA and schedules content matched to these preferences.

Social Capital View – Figure 4(b) This operation mode contains all information relating to the presented App. If a user LSA with application-specific data is inserted into the public display's local LSA-space, this view is centered and updated to display this data and the generated recommendations. Data for two users can be visualized at a time. Steering information – guiding the user to the computed food provider – is shown and the current SCI and its converted value are visualized. This view further contains a representation of the user's utility distribution (mapping behaviors to credit rewards), the ratios of the behavior components (see subsection IV-A) and of the conversion function's development over time.

The presented App running on the private displays allows input of food preferences, individual weights of components for the computation mechanism of the optimal behavior and a visualization of the LSA-space. The various views are referenced in Figure 5.

C. SAPERE development experience report

Implementing complex applications using the SAPERE paradigm and middleware offers multiple advantages over established technologies such as client-server models or aspect-oriented programming. On the one hand, SAPERE naturally supports decentralized systems with a massive number of interacting components that are self-coordinating and adaptive in relation to local context as well as to the overall system structure. Indeed the middleware takes care of promoting spontaneous communication between nodes and starts the execution of applications. On the other hand, the four eco-laws proved sufficient to coordinate SAPERE components effectively and to implement complex interaction patterns, distributed sensor applications and context-triggered interactions between display devices.

VI. Related Work

This section provides reference works for pervasive models and middleware, and existing background on applications involving social feedback.

A. Pervasive Models and Middleware

From a general perspective many works are related to SAPERE. In this section we focus in particular on situatedness, context-awareness and self-organization.

The idea of situatedness has been promoted by the adoption of shared virtual spaces for services and component interactions, introduced first by *Gaia* [9]. The *Gaia* infrastructure [9] introduces the concept of active spaces for pervasive collaboration environments. Similar concepts have been exploited for programmable tuple spaces. *Lime* [10] is a middleware that provides support to both physical and logical mobility. In *Lime* the concept of a global tuple space disappears, each agent keeps a private tuple space. When agents are within range, their privately owned tuple spaces can merge together to be used as a

common interaction space. *EgoSpaces* [11] is a middleware for ad hoc mobile environments focused around the abstraction of a view, that is an agent-specific notion of context. *TOTA* [12] is a tuple-based middleware supporting field-based coordination for pervasive-computing applications. In *TOTA* the focus is on the tuples that embed rules for their own replication over a network of P2P nodes. While exploiting tuple spaces and a number of common features, SAPERE has a different space concept than all the cited approaches, indeed local spaces in SAPERE cannot merge – as the ones in *Lime* do. The behaviors are embedded in the eco-laws rather than in the tuples – as in *TOTA*. Moreover agents do not need to declare a personalized view of the context as in *Ego-spaces*, they simple notify their existence by injecting their own LSAs.

Several works exploit the properties of adaptive self-organizing natural and social systems to enforce self-* behavior in pervasive computing systems. *Anthill* [13] is a framework built to support design and development of adaptive peer-to-peer applications, that exploits an analogy with biological adaptive systems. SAPERE has the more ambitious goal of fully supporting adaptive coordination in distributed applications rather than building robust and adaptive networks of peer-to-peer services. *SwarmLinda* [14] is a middleware inspired by the collective intelligence displayed by swarms of ants. It acts as a single and globally accessible tuple space and the self-organizing processes take place as internal mechanism of the middleware. In SAPERE we rely on a different biological metaphor, however with *SwarmLinda* we share the idea of having self-organizing mechanisms inside the middleware that are completely transparent to the entities.

B. Social Feedback Models

In this work, we adapt a number of well established behavioral models [15], [16], [17] – that already provide a measure of a user's behavior and quantify society's reaction to this behavior – to create a weighted, multi-component representation of such user behaviors. We further evaluate them in regard to their value to society based on social feedback theory established by Pierre Bourdieu [6] and expanded upon by [18], [19]. Using these abstractions, we build a dynamic, spontaneously adapting social feedback application for a pervasive service ecosystem.

Inspired by similar projects [1], [2], [3], we believe public/private display technologies are an effective and realistic platform to implement such an application.

VII. Conclusion

This paper described an instantiation of a generic social feedback application above the infrastructure of a pervasive display ecosystem exploiting the SAPERE middleware. We introduced a social utility function indicating to what extent a behavior benefits the overall system and its components, and recommended behavior to users based on weighted versions of the user's preferences, the behavior's value to "society" and the quantified effort to adopt a behavior. We illustrated the actual implementation of the system to give a better understanding of the application purpose and mode of operation.

Acknowledgment

Work supported by the SAPERE (Self- Aware Pervasive Service Ecosystems) project (EU FP7-FET, Contract No. 256873).

References

[1] F. Gollan, B. Wally, and A. Ferscha, "Automatic attention estimation in an interactive system based on behaviour analysis," in *Proceedings of the 15th Portuguese Conference on Artificial Intelligence*, 2011.

[2] N. Pears, D. Jackson, and P. Olivier, "Smart phone interaction with registered displays," *IEEE Pervasive Computing*, vol. 8, no. 2, pp. 14–21, 2009.

[3] T. Ojala, H. Kukka, T. Lindén, T. Heikkinen, M. Jurmu, S. Hosio, and F. Kruger, "Ubi-hotspot 1.0: Large-scale long-term deployment of interactive public displays in a city center," in *Proceedings of the 2010 Fifth International Conference on Internet and Web Applications and Services*. IEEE Computer Society, 2010, pp. 285–294.

[4] A. Di Ferdinando, A. Rosi, R. Lent, A. Manzalini, and F. Zambonelli, "Myads: A system for adaptive pervasive advertisements," *Pervasive Mob. Comput.*, vol. 5, no. 5, pp. 385–401, 2009.

[5] A. Rosi, A. Codeluppi, and F. Zambonelli, "Adaptive pervasive advertisement: scenarios and strategies," *International Journal of Pervasive Computing and Communications*, vol. 6, no. 3, pp. 333–351, 2010.

[6] P. Bourdieu, *The Forms of Capital*. New York: Greenwood Press, 1986.

[7] F. Zambonelli, G. Castelli, L. Ferrari, M. Mamei, A. Rosi, G. D. M. Serugendo, M. Risoldi, A.-E. Tchao, S. Dobson, G. Stevenson, J. Ye, E. Nardini, A. Omicini, S. Montagna, M. Viroli, A. Ferscha, S. Maschek, and B. Wally, "Self-aware pervasive service ecosystems," *Procedia CS*, pp. 197–199, 2011.

[8] F. Zambonelli, G. Castelli, M. Mamei, and A. Rosi, "Integrating pervasive middleware with social networks in sapere," in *International Conference on Selected Topics in Mobile and Wireless Networking*, 2011, pp. 145 –150.

[9] M. Román, C. K. Hess, R. Cerqueira, A. Ranganathan, R. H. Campbell, and K. Nahrstedt, "Gaia: a middleware platform for active spaces," *Mobile Computing and Communications Review*, vol. 6, no. 4, pp. 65–67, 2002.

[10] A. L. Murphy, G. P. Picco, and G.-C. Roman, "Lime: A coordination model and middleware supporting mobility of hosts and agents," *ACM Trans. on Software Engineering and Methodology*, vol. 15, no. 3, pp. 279–328, 2006.

[11] C. Julien and G.-C. Roman, "Egospaces: Facilitating rapid development of context-aware mobile applications," *IEEE Trans. Software Eng.*, pp. 281–298, 2006.

[12] M. Mamei and F. Zambonelli, "Programming pervasive and mobile computing applications: The tota approach," *ACM Trans. Softw. Eng. Methodol.*, vol. 18, no. 4, pp. 1–56, 2009.

[13] O. Babaoglu, H. Meling, and A. Montresor, "Anthill: A framework for the development of agent-based peer-to-peer systems," in *Proceedings of the 22nd International Conference on Distributed Computing Systems*. IEEE Computer Society, 2002, p. 15.

[14] R. Tolksdorf and R. Menezes, "Using swarm intelligence in linda systems," *Lecture Notes in Computer Science*, pp. 519–519, 2004.

[15] B. J. Fogg, "A behavior model for persuasive design," in *Proceedings of the 4th International Conference on Persuasive Technology*. ACM, 2009, pp. 1–7.

[16] M. Fishbein and I. Ajzen, *Belief, attitude, intention, and behavior: an introduction to theory and research*, ser. Addison-Wesley series in social psychology. Addison-Wesley Pub. Co., 1975.

[17] V. Vroom, *Work and motivation*. Wiley, 1964.

[18] M. Siisiäinen, "Two concepts of social capital: Bourdieu vs. putnam," in *ISTR Fourth International Conference, The Third Sector: For What and for Whom?*, 2000.

[19] P. Dekker and E. M. Uslaner, *Social Capital and Participation in Everyday Life*. London: Routledge, 2001.