Self-Organizing Data Sharing Communities with Sagres

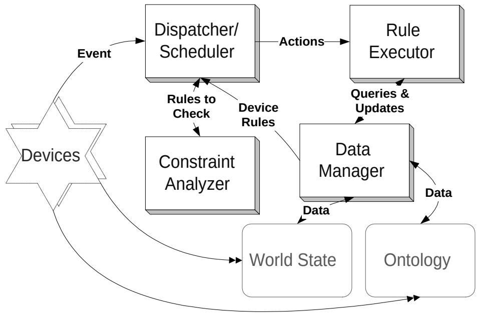
Zachary Ives Alon Levy Jayant Madhavan Rachel Pottinger
Ştefan Şaroiu Igor Tatarinov Shiori Betzler Qiong Chen Ewa Jaslikowska
Jing Su Wai Tak Theodora Yeung

University of Washington
{zives,alon,jayant,rap,tzoompy,igor,shiorith,qiong,emjeden,jingsu,theodora}@cs.washington.edu

1 Introduction

An increasing number of devices (e.g., household appliances, PDAs, cell phones) have microprocessors and will soon be able to exhibit sophisticated behaviors and interactions with other devices: a home heating system will monitor its residents' alarm clocks and schedules to set the temperature optimally; a car's GPS system will use local traffic reports to optimize its driver's route based on road conditions. The Sagres project at the University of Washington addresses the key issues of data sharing and management in the realm of invisible computing.

In the context of invisible computing, data exchange and computation occur in the background in response to cues from users. Devices are added and removed from the network on a regular basis, and they must be able to interoperate with little human intervention. The collection of devices that exist around a particular individual or in a particular location (e.g., a house, office building or virtual networked location) form a *data sharing community*. Ultimately, these devices must share data in a common format such as XML. Managing device interaction and the flow of such data is key to the operation of data sharing communities. We demonstrate the Sagres system, which represents the first step in a large-scale project. Our ultimate goal is to develop a new programming paradigm and a scalable architecture for developing self-configuring devices under various user constraints and preferences. Issues we tackle include development of both a language and an environment for managing distributed, resource-sharing, concurrent device interactions and transactions.



2 The Sagres Architecture

Device functionalities and data must be represented within Sagres (Fig. above), but the key to the system lies in our novel programming model. The rule-based DevL language is event-driven, and combines semistructured data manipulation capabilities with constraints, synchronization, and message passing to facilitate seamless device interactions.

Sagres data is stored in two graph structures: an Ontology, which represents the class/inheritance hierarchy of entities controlled by Sagres, as well as rules, queries, and attributes; and a World State, which is a virtual view of the data present in the data sharing community.

As a device joins the system, it adds its class information to the Ontology and its data to the World State. Next, rules for the new device are analyzed by the Constraint Analyzer to verify they do not conflict with existing actions or constraints.

On an event, the Dispatcher looks up matching rules, tests their preconditions, and sends their actions to the Executor. The Executor performs the operations, querying and updating the World State and Ontology through a Data Manager. (Updates may initiate device actions.)

The vision of invisible computing encompasses nearly all subfields in computer science, and will require contributions from all of these communities to become a reality. Sagres attempts to address the related data management issues, thereby representing the database community's contribution to this emerging area.