A Plugin Framework to Control Electronic Shelf Labels

Gerrit Kahl
German Reasearch Center for
Artificial Intelligence
Campus D3_4
66123, Saarbrücken, Germany
gerrit.kahl@dfki.de

http://dx.doi.org/10.1145/2494091.2496014

Abstract
Nowadays, more and more electronic displays are integrated into modern supermarkets in order to display advertisement or to act as price tags referred to as electronic shelf labels (ESL). Due to the low power consumption, wireless accessibility and appropriate resolution, these displays represent an alternative to state-of-the-art paper printed price labels. Nevertheless, there are several types of displays regarding size, resolution and communication channel. In this paper, we present a framework to automatically generate display content for different display types. Besides the connection to the displays, the framework also includes the possibility to generate the content to be displayed based on application plugins. The plugin approach facilitates both the integration of new display types and the creation of new applications. We explain the prototypical implementation of several plugins as well as the application at an instrumented shopping environment.

Author Keywords
Plugin Framework, Management of Electronic Shelf Labels, Instrumented Supermarkets

ACM Classification Keywords
H.5.2 [Information interfaces and presentation (e.g., HCI)]: User Interfaces.

General Terms

Architecture Design, Prototypical Implementation

Introduction

Today used paper printed price labels have the advantage of being cheap and well known by customers. If prices change, however, new labels have to be printed and placed at the correct shelf by the supermarket staff. For example for temporary price drops, such as weekly special offers, as well as if new products are added to the sales mix, new price tags have to be printed which implies costs and keep staff from assisting customers or refilling shelves implying a loss of revenue. A further disadvantage of those printed price tags is that it might happen that there are wrongly placed labels or some not displaying the actual price, which could lead to upset customers. Studies by Lach and Tsiddon [6] have shown that price changes occur in fact very frequently in common retail environments. Dutta et al. [1] did cost calculation based on both, the labour and printing cost of price changes, as well as on the costs for mistakes made during the price change process and the supervision of the process to avoid those mistakes. In 1996, Levy et al. [7] calculated a sum of these costs of approximately 100.000 $ per year for a U.S. supermarket store with 25.000 products.

Regarding this sum, the costs for changing to electronic displays will amortize within a few years. Despite that, modern ESL offers the possibly to display price information in the same resolution and layout as paper printed versions and they can be controlled from a central location due to wireless communication. These reasons lead to a substitution of paper printed price labels for several markets[1]. Depending on the communication channel and resolution of the displays, specific software toolkits - often also provided by the companies selling the ESL systems - are needed to create and to send content to the labels. Apart from that, ESLs offer the possibility to display more information than just prices. E.g. a store manager could think of additional advertisement, attention guiding for customers or any other application that would be a benefit for the store. Besides that, the acceptance of interactive technologies at the point of sale has already been proven by Loebbecke [8]. This leads to the conclusion that such an ESL infrastructure could be used in user assistance systems. In this paper, we present a framework for content generation and data transmission to several kinds of display types based on a plugin approach.

[1] for example http://www.bizcommunity.com/Article/196/412/92376.html

ACES

We have developed a framework for automatic generation and transmission of content for ESLs called *ACES* - **A***pplication and* **C***ontrolling Framework for* **E***lectronic* **S***helf Labels*. Based on a plugin approach, several display types as well as a variety of applications can be supported by ACES, while the applications can react on changes gathered by sensors in an instrumented environment. The core of the framework consists of the controller, the display manager, the application manager, and the datasource manager. Figure 1 illustrates the communication flow within the core as well as the connection to external services. In the following, we describe each component in detail starting with an external service that is used for the communication between ACES and the instrumented environment.

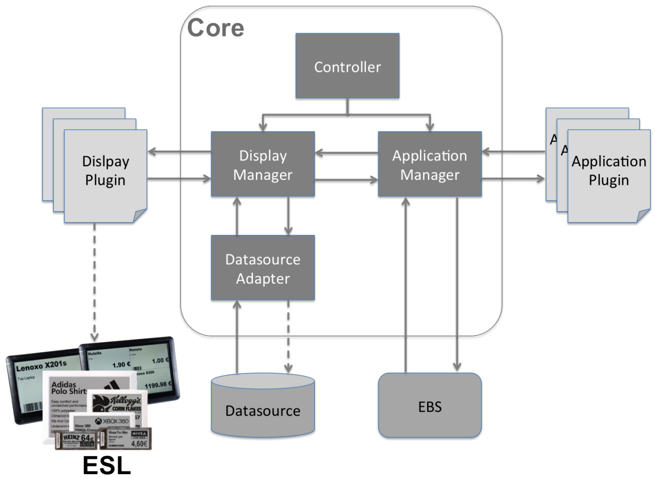

Figure 1: Architecture of ACES

EBS
In order to trigger a content change of an ESL, the Event Broadcasting Service (EBS) is used [3]. The EBS is an event-based communication infrastructure for smart environments. All components of the environment can communicate with each other by encoding the information into events, which are sent to a central server. This server broadcasts the event to all connected clients, which can process the information. ACES registers to the EBS and retrieves all data gathered by sensors installed in the environment. For example, price changes and object as well as user location information can be sent as events using the EBS infrastructure and processed by ACES.

Controller
The *Controller* acts as the starting point of the framework and manages the further components. It parses configuration files including the plugins to be used. Due to the usage of the Java Reflection API [2] new display and application plugins can be added at runtime without restarting the system. For managing the used plugins, a web interfaces is provided by the controller.

Datasource Adapter
The *Datasource Adapter* manages the communication to the datasource. By implementing several datasource plugins, information can be taken from a standard database, a merchandise information system, or even an XML file. The datasource adapter itself acts as a wrapper around this datasource plugins while either caching the information when the system starts or holding a connection to the datasource. It provides lists of all displays placed in the supermarket, products, and shelves including the geometric information, such as positions and orientations. Using the datasource adapter, connections between products and displays can be determined as well as environmental information, such as directions from one display to a product within the store.

Application Manager
The *Application Manager* organizes the connection to the EBS. Whenever a new event is received it is forwarded to all registered application plugins. These plugins encompass the functionality of the associated applications, which comprise the filtering of events they are interested in and the generation of the content for the ESLs. Missing information that is not included in the event can be obtained from the datasource adapter. Using the connection to the display manager, the application manager provides the corresponding display information, such as the resolution, which is used in the content creation process.

Display Manager
The *Display Manager* holds a list of all displays connected to the framework and manages the information

Figure 2: Image of a price label.

transmission to them. The connection itself is realized using display plugins. For each display type, a corresponding plugin has to be implemented, which includes both the physical information of the display (e.g. size, color information) and the communication channel (e.g. Bluetooth, WLAN). In order to ensure that only one application can make use of a display at the same time, the display manager provides a locking mechanism. This locking mechanism is decentralized by issuing tickets to the application plugin. These tickets can also be invalidated by the display manager in order to prevent deadlocks. In addition to the communication to the displays, the display manager is also able to alter the underlying datasource by communicating with the datasource adapter.

If an application plugin wants to connect to a display, this request is forwarded by the Application Manager to the Display Manager, which checks if the display is used by another application. If it is free, a new ticket is granted and the corresponding display plugin is forwarded to the application plugin. The display plugins can also store a standard screen content, which is sent to the displays if specific screen content should only be displayed temporary.

Prototypical Implementation

We have integrated the ACES framework in an already existing research lab that focuses on the support of customers and staff with visionary assistance systems for the retail domain.

IRL

The *Innovative Retail Laboratory (IRL)* is an application oriented living lab [9]. In this living lab, which is set up like a supermarket, new assistance systems for the retail domain are developed and tested. The focus is not only to assist customers during their shopping process but also to support the employees of the supermarket in their business. One service helping both stakeholders is the usage of ESLs as described in the introduction. While the employees do not have to consider about the pricing and replacement of labels, customers get a clear overview of the prices that are synchronized with the merchandise information system, which reduces the problem of wrong pricing information.

At the IRL we use shelf labels provided by ZBD solutions[2]. They provide a complete ESL support including return on investment (ROI) analyses, software for content creation and transmission, and the hardware referred to as *epop* (electronic point of purchase). There are several types of epops offering different resolution and size, but all can be accessed wirelessly using a specific communication device, the *Bounce Communicator*. Content creation and epop updates can be automated based on pre-defined templates triggered by database changes. In addition, ZBD also offers an API including an interface to send existing images to the epops.

Application Plugins

Prototypically, we have implemented several applications including dynamic pricing, navigation functionality and a game using the ACES framework. Besides the triggering based on events received from the EBS server, the application plugins also enable an internal trigger in order to perform actions periodically, e.g. to display advertisement at the ESLs. In order to trigger changes in the environment based on new created label content, the plugins can also send events to the EBS using the interface provided by the application manager, e.g. to

[2] http://www.zbdsolutions.com

Figure 3: Arrow guiding a customer in the micro navigation mode.

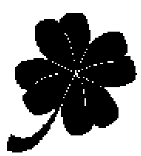

Figure 4: Clover identifies products to be scanned.

Figure 5: Multiscreen display.

control a steerable projector for displaying a spot at a specific product while the ESL is adapted simultaneously.

Price Setter

One of the most obvious applications for ESLs is the presentation of product information including prices. Whenever a "price change event" is sent by a service of the instrumented environment, it is processed in the price setter plugin. The product's European Article Number (EAN) included in the event is used to generate a barcode. In addition price, producer and a short description including the net weight are used for the creation. According to a predefined template and the resolution of the display belonging to the product, the image is generated (see Figure 2).

Micro Navigation

The micro navigation is used to guide a customer on shelf meter level to a specific product he is looking for. Therefore, arrows are displayed pointing to the searched product using the location of both the product and the display (see figure 3). The micro navigation is not only changing the content of one ESL but of all containing to the shelf. When the product is found by the customer, which is identified by the sensors of the environment and communicated by the EBS, the standard content is sent to the displays. In order to start this plugin, information about the customer's current location is needed. This information is provided by an instrumented shopping cart, the IRL SmartCart [5]. The cart can estimate its location by recognizing tags that are placed under the flooring of the IRL using an RFID antenna mounted in between its wheels and sends the estimation via the EBS.

Clover Game

The ACES framework has also been integrated into an Android game where customers can gain a discount on their purchase when they scan specific products within the store. Products to scan can be specified by the shop owner. When a user is nearby a product he shall scan - determined by localization based on scanned products or on the positioning system integrated in the IRL SmartCart - the corresponding ESL displays a clover until the user goes further away or scans the product (see figure 4), while positioning information is transmitted as events to ACES from the sensors and services.

Display Plugins

Despite from electronic shelf labels based on ePaper and wireless control mechanisms, we have also implemented plugins for further display types. Due to power consumption, updates for standard ESLs are quite slow. Displays connected to a computer or even smartphones, however, have the advantage that they can react immediately on changes.

Fullscreen Plugin

The first plugin makes use of a screen connected to a computer. For example monitors connected via USB such as MimoMonitors[3] can be used as ESL. Additionally to the displays connected to the computer running the ACES framework, displays connected to further computers can also be used using Java RMI.

Multiscreen Plugin

Using large displays, such as monitors, it might be helpful to split the screen in order to use the display for more than one product. For this purpose, we have implemented a multiscreen plugin dividing the screen into four parts. Each can be accessed as a separate ESL (see Figure 5).

[3] www.mimomonitors.com

EPOP Plugin
As described above, we use ESLs provided by ZBD solutions. The implemented epop plugin comprises the different epop types and manages the communication using the Bounce Connector.

Exemplary Run
In order to change a price of a product, a price changed event can be sent by any component of the environment. This event comprises a reference to the object which price should be changed as well as the new price. It is broadcasted by the server to all connected clients including ACES. In ACES, the object reference is used to identify the corresponding display and to obtain all information about the product from the datasource adapter. Concurrently, the datasource adapter updates the merchandise information system. Based on the event type, the price setter plugin is triggered, which creates a corresponding images according to the template including the new price. Afterwards, the plugin requests a ticket for the corresponding display, which is granted by the display manager. Finally, the display plugin transfers this image to the display according to the defined communication channel.

Discussion
In this paper, we have presented a framework for content creation and management of electronic shelf labels in supermarkets. It allows the assembly of a very heterogeneous network of ESLs. The proposed plugin approach enables the usage of different display types as well as several applications. The framework has prototypically been implemented in a retail laboratory. In this context several display as well as application plugins have been developed and implemented.

For the future, we think about implementing more user adaptive application plugins. For example displaying special prices for a specific customers or using the ESLs for group advertisement. The framework could also be used in an interactive scenario where the user can interact with the ESL either direct or indirect, e.g. for price negotiation. Dynamic pricing as described in [4] is also a promising topic using an ESL infrastructure. For example, the prices can be automatically adapted to the best before dates of the products. When using monitors for displaying prices, the implementation of a new application plugin also allows to play videos. This can be used to display product recommendations, for example.

In the current version of ACES, we do not handle synchronous access to a display. Whenever an application plugin requests a display which is not in use by another plugin, it can access it. Otherwise, it waits until the display is free and sends the content then. When displaying videos, ESLs will be occupied for a long time, which prevents them to display pricing information. Therefore, new mechanisms to resolve those collisions have to be integrated into ACES.

Acknowledgements
We thank David Kunzler for his ideas, commitment and implementation for the prototype.

References
[1] Dutta, S., Bergen, M., Levy, D., and Venable, R. Menu costs, posted prices, and multiproduct retailers. Tech. rep., EconWPA, May 2005.
[2] Forman, I. R., and Forman, N. *Java Reflection in Action (In Action series)*. Manning Publications Co., Greenwich, CT, USA, 2004.

[3] Kahl, G., and Bürckert, C. Architecture to Enable Dual Reality for Smart Environments. In *Proceedings of the 2012 Eighth International Conference on Intelligent Environments*, IE '12, IEEE Computer Society (Washington, DC, USA, 2012), 42–49.

[4] Kahl, G., Magerkurth, C., Preißinger, J., Gebhard, P., and Weyl, B. Enhancement of consumer support in retail scenarios by utilization of semantic product memories. In *SemProM: Foundations of Semantic Product Memories for the Internet of Things*. 2013, 329–347.

[5] Kahl, G., Spassova, L., Schöning, J., Gehring, S., and Krüger, A. IRL SmartCart - A User-Adaptive Context-Aware Interface for Shopping Assistance. In *Proceeding of the 15th international conference on Intelligent User Interfaces. International Conference on Intelligent User Interfaces (IUI-11), February 13-16, Palo Alto,, CA, United States*, ACM (2011).

[6] Lach, S., and Tsiddon, D. Staggering and synchronization in price-setting: Evidence from multipro-duct firms. Working paper, National Bureau of Economic Research, June 1994.

[7] Levy, D., Bergen, M., Dutta, S., and Venable, R. The magnitude of menu costs: Direct evidence from large u.s. supermarket chains. Macroeconomics, EconWPA, 2005.

[8] Loebbecke, and Claudia. Use of innovative content integration information technology at the point of sale. *European Journal of Information Systems 16*, 3 (July 2007), 228–236.

[9] Spassova, L., Schöning, J., Kahl, G., and Krüger, A. Innovative Retail Laboratory. In *Roots for the Future of Ambient Intelligence. European Conference on Ambient Intelligence (AmI-2009), oA, Salzburg, Austria (November 2009)* (2009).