Smart Shelf: Report

Md. Abdul Kadir
Saarland University
Saarbrücken, Germany
maktareq@gmail.com

Kevin Denk
Saarland University
Saarbrücken, Germany
denk.kevin@web.de

Atika Akmal
Saarland University
Saarbrücken, Germany
atikaakmal19@gmail.com

ABSTRACT

We are on the apex of interaction technology integration with every analogue system that is used in our daily life. Working in big lab with lot of consumable equipments or finding a product in a supermarket is really a big deal. It's really hard for a person to find an object from a gigantic shelf. Also, it's hard to keep supply continuous of consumables by checking every individual slots. Sometime, it's impossible. To ease the hard work of management and user we introduce *Smart Shelf* that will bring smoothest interaction between shelf and human. Here, we present the overall framework, related work and the time line to implement a prototype of *Smart Shelf*.

ACM Classification Keywords

H.5.m. Information Interfaces and Presentation : Human Computer Interaction

Author Keywords

Smart Fabrication, Smart, Shelf, Drawer, HCI, Human Computer Interaction, Physical Computing

INTRODUCTION

Today the word *Smart* is almost everywhere. There are *Smart Homes* and *Smart Fabrication*. But from the last few years, HCI researcher have developed new interactive interfaces, synthesize from different perspective of humans for example Psychology, Economically and Social that can be integrated with modern technologies (Ubiquitous technology). People prefer to use these technologies but sometimes face problems to obtained desired results. For example, most of the time people don't like to use shelves, because it is hard to find something. By using ubiquities technology, combined with technologies of the *Smart Home* and *Smart Fabrication* domain an interactive shelf, called *Smart Shelf*, can be developed. This shelf can enhance the utility of regular shelves. This report describes the approach of the implementation of the *Smart Shelf*. The report discuss different approaches of the interactive design, how the *Smart Shelf* is implemented and how to interact with it. One focus is the interaction of users and operators with the shelf. In the end there is an evaluation of the design decisions regarding the project and an outlook for further work.

Problem

Most people when they hear about a shelf they think about their bookshelf or some shelves in the kitchen. Almost everyone who have a bookshelf searched at least one time in his/her life for a book in it and wished to have a guideline how to find it the fastest way. Imagine big shelves with a lot of small drawers. Every drawer is only labelled with a small name that describes what is in that drawer. Searching for items in these shelves can be hard and cost a lot of time. An additional scenario is if you apply this concept to big warehouses with hundreds of shelves and more drawers or places where you can place items. Finding an item in such a warehouse can be still harder.

Shelves used in companies or research institutes bring more problems to the surface. Often there are shelves used for storage of electronic components [1] for example. The drawers contained in these shelves are often a lot and small. Searching for a specific item/drawer in such a shelf with for example fifty drawers can be exhausting. But this is not only one problem. Is the searched drawer found, the contained items are maybe out of stock and the user wasted his/her time. This is not only exhausting, it is also wast of time if the user could directly see if the searched item is out of stock.

Not only warehouses or storage rooms with shelves have those problems with the inefficiency in finding items or the premise if one item is out of stock. The same problems appear in retail. Customer which can't find their favourite product in a shop are unsatisfied. Maybe they go to another shop and don't come back. This problem can be tackled with *Smart Shelves*, too. The shelf itself could detect if some products in it are only available in a small amount. In this case the shelf could order new products or at least send an information to an operator who can order supplies. With this strategy there will be no more empty shelves in shops and customers can find their favourite product all the time.

Smart Shelf should be a solution for these problems. It could observe the amount of items in itself, help people to find products and also order supplies if the amount of items is low. The project described by this report tries to give a solution for the mentioned problems.

RELATED WORK

There are several development work happened last few year in human computer interaction(HCI), home automation and embedded technology. A big set of these work is giving intelligence to rigid objects and allow human to communicate

---

[1] e.g. Resistors, Capacitor, Micro Controllers or Integrated Circuits

with them and vice-versa by applying noble HCI techniques. Moreover, post-WIMP devices also offer some features that can be integrate with the modern computer technology development(Ubiquitous computing). However, this post-WIMP GUI concept only applicable if there is a metaphor available in digital or analogue world. For example, searching the meaning of a word in digital dictionary(e.g:Smart phone dictionary). We want explain decent amount of successful research work that overlap at least in certain area with our Smart Shelf framework; However, there is no implementation or ground work fully overlap with our concept. A technical definition of our project is "Combining different interaction technique to innovate a device that follow the guideline of ubiquitous computing".The most related topic that are already known by design community are: QR code for presenting information, Automatic amount calculation, Controlling device.

QR code for presenting information

Now a days application of QR code become very popular and common due to the smart phone technology. Now people don't need to type search. Pressing a key is enough to get information based on QR code. A very innovative application is using QR code in library management. In a case study ' 'Application of QR Code Technology in providing Library and Information Services in Academic Libraries" by Sandeep Kumar Pathak showed that important information can be presented by QR code and user can easily get all those information by scanning QR code. We are implying this idea in to completely different perspective. In our case every drawer will have individual QR code. Each code will represent individual information about items stored in the drawer.

Automatic amount calculation

One major objective our implementation is representing empty or not empty drawer. As it's a very ground level work of many automation project, there are many project information available regarding weight measurement. However, in our project we are counting the objects based on the overall weight. We don't see this sort of work is not very common to automation community.Although, the most related work to that sub task is counting weight based on resistive sensor. An example of this work presented in *circuitdigest.com*:Arduino Weight Measurement using Load Cell and HX711 Module. Here they use Load Cell, but we will use resistive load sensor to calculate the weight signal.Also, their project does not include counting.

Controlling Device

The most important human to machine interaction task in this project is giving command to the system. There are several way to build up the this interaction system:one could be developing from scratch and another is building up over existing individual system. It's very common in Internet of Things (IoT) community to to use a smart phone for controlling a electronic system.For example there are lot of projects that use android devices for home automation. However, Our approach is similar but objective is completely different.

Figure 1. Example for used Shelf as prototype

We see there are many existing work happened in granular level, but here we are bringing these granular ideas to build completely a new noble system.

CONCEPT

This section describes the basic concept of the *Smart Shelf*. The basic idea of the Smart Shelf is to enhance the interaction with shelves to work with these more effective and to improve the interaction. To satisfy these objectives some core features are proposed in this section. Figure 1 shows a similar shelf used for the project prototype which is small but satisfies the requirements for this prototype. However, one important fact which should be considered is the scalability. The applied techniques should be feasible in a technical and economical way.

The first problem the Smart Shelf wants to address is the fact that it is hard to find a specific item in a high amount of drawers. To do this we propose visual feedbacks combined with an user interface to submit search queries. The user interface can be used to insert a search word, for example the name of the searched item. This search query results in a visual feedback on the Smart Shelf, or in detail on the specific drawer where the item is contained.

The user interface will be provided as webapp. The platform independence of a webapp leads to this design decision. In a first iteration mobile apps are considered, but were discarded because of the platform dependency (Android, Apple iOS, Microsoft Windows, etc.). Additionally, more features are provided by the webapp. Overall, it provides all functionalities to interact with the Smart Shelf. The search, control the shelf or getting more data about different items. For getting more information about a specific item the user interface provides additionally to a datasheet-like page the capability to scan QR-Codes. These QR-Codes are mounted on each drawer

and encode the id of the drawer. A central server, which also serves the webapp holds the state of the Smart Shelf and has a mapping of these scanned ids to drawers and items. The different items contained in the shelf, the amount of them and current searches are called here as *state*.

As mentioned before an unsatisfying fact is, if the user search for a while for a specific item and finally find it, the drawer is empty. To address this problem the Smart Shelf provides a *predictive management system* (abbr. PMS). This PMS enables the system to send notifications to operators if a drawer is empty or almost empty. Notifications enables the operators to react early to items with low or zero amount. This decreases the chance of out of stock items. To realize such a PMS data are needed for analysis and reaction based on specific conditions. Every drawer will get a electronic unit to measure the weights of the items in a drawer. Theses measurements are send to the server, which calculates with the stored weight of the specific item, the amount of items in that drawer. This enables the PMS to have an overview of the state about all items contained in drawers. Additionally to notifications a visual feedback is given at the shelf itself. Empty drawers are marked with a red light to indicate this to users directly.

As mentioned before visual feedback is a central point in the design of the Smart Shelf. Therefore, it is used to indicate the searched drawer to the user and if a drawer is empty. However, the prototype uses visual feedback additionally for a so called *service mode*, which can be switched on and off. During service mode every drawer visualizes its state with different colours. Blue colour, if the drawer contains more items than a specified threshold. Yellow colour as warning, if there are less items than the threshold and a red colour, if there are zero items left. For this service mode the data acquired with the weight measurement are consulted.

Visual Feedback Colours

Assumptions
To realize this prototype some preconditions are assumed which are listed in this section.

All items in the shelf are either without packaging or with packaging. There will be no drawer where items are mixed. Furthermore, at any time there is only one sort of items in one drawer. Every drawer contains different items. The weight of a specific item is stored in the server for the amount calculation.

IMPLEMENTATION
This chapter shows the important details of the Smart Shelf implementation. The system of the Smart Shelf itself consists of three main components. The backend server with a MySQL database, a webapp served by the backend and Smart Shelf itself with an Arduino as main microcontroller connected to some electronic components placed in the drawers. Basically, the Smart Shelf is contained of several Smart Drawers. These components and how they communicate with each other are described in the following.

Backend
For the backend implementation the Spring framework is used. This enables to build a lightweight backend server that can serve web pages, a REST API and can connected to a MQTT[2] broker. The communication between backend and Arduino is explained in section 4.2 in detail. To serve the webapp the backend implements several REST Services which can be consumed via HTTP. Some of this services produce the different web pages for the webapp. Others produce just data to keep the information in the webapp up to date.

As mentioned before the backend keeps the state of the whole system. For that it implements some queues for all current search queries and has a connection to a MySQL database with all the essential data needed for the Smart Shelf. The database contains for example all items contained in the shelf, in which drawer these items are and what boxes are there. Additional information like the amount of different items are stored, too. To enable the PMS the backend implements also some services to mark an item as out of stock and to send an email to defined operators as notification. These operators can subsequently order supply for the appropriate items. In addition to the sent notification the backend sends a command to the Arduino to switch on the red LED on that drawer. This signals directly to each user that this drawer is empty.

To use more of the capabilities the visual feedback provided via the LEDs on the drawers the Smart Shelf implements a so called service mode. This mode can be started from the webapp. After started the backend checks the state of all drawers and send the appropriated commands to the Arduino to switch on the correct LED on all drawers. As mentioned before there is a mapping between the colours of the LED and the amount of items in a drawer. Blue colour signals that the amount of items in that drawer is above a defined threshold. Yellow colour indicates that the amount is below the threshold and red visualizes that this drawer is empty.

Communication
For the communication between Arduino and backend the lightweight M2M[3] communication protocol MQTT was chosen. MQTT works with the publish and subscribe pattern. There is a central broker where clients can subscribe topics. Additionally, clients can publish messages to different topics. If a client subscribed the topic under which a message is published, the client will receive it [1]. For further information about this protocol check [1].

With MQTT a bidirectional connection can be established between backend and Arduino. Primary, the backend send commands to the Arduino which LEDs the Arduino should switch on or off. But additionally, the Arduino send in a defined interval the measured weights of items contained in different drawers. How the weight measurement is done is explained in following chapters.

---

[2]Abbr. Message Queuing Telemetry Transport Protocol
[3]Abbr. Machine-to-Machine

Figure 2. Mainpage fo the webapp

Smart Drawer

Webapp

To control the Smart Shelf a webapp is served by the backend. This webapp can be opened via an URL from almost every browser which is connected to the network the backend is in. Implementing the user interface as webapp brings the advantage that there is no platform dependency. It can be used form a Android or iOS phone as well as with computer.

Figure 2 shows the main page of the user interface. The main function of the system is to make it easy searching for a item in the shelf. Therefore, a big search field can directly be seen at the main page. Inserting there the name of the searched item results in a list of items matching the inserted name. Additionally, all LEDs of the drawers which contain the displayed items in the webpage will be turned on. By using the SSelect-button for the item to choice all LEDs will be switched off except the selected one. With this functionality the user can easily find the search item without scanning the all drawers for the specific name.

To get more information about an item one can uses the data sheet provided by the webapp. This can directly reached from the search result, also seen in figure 2, or by scanning the QR-Code mounted on the appropriate drawer. The QR-Code brings the advantage that if a user need more information about a specific item and is standing right in front of the shelf he/she doesn't have to search again and afterwards select the data sheet. The data sheet can easily get by scanning the QR-Code with the smart phone which used to operate the webapp. As mentioned before the QR-Code encodes the ID of the drawer. By scanning this ID is decoded, send to the backend which responds with the needed information about that drawer to visualize the correct data sheet.

EVALUATION

bla bla fun

FUTURE WORK

CONCLUSION

We have deployed the content of physical computing in the form of interactive smart shelf where user interact via web app and get visual feedback according to operation. Our main goal to design interactive smart shelf to reduce conscious thought while search an item as well as provide reliable and efficient way of searching , save time and provide multiple user interface. By using Ubiquitous and other technologies, interactive smart shelf enhance the usability of shelves.

REFERENCES

1. OASIS Standard. 2014. MQTT version 3.1. 1. *URL http://docs. oasis-open. org/mqtt/mqtt/v3* 1 (2014).