

Clever Laundry Room

Stefan Negritoiu

stefann@stefann.com

1. Background

Technology advancements in today's world allow us to design better solutions faster, easier and cheaper than ever before. The price of employing software and hardware in solving real world problems used to be so high that the activity was confined to only a few economic sectors whose return would match such prohibitive costs.

Today we have personal computers whose computing power exceeds basic needs of common users. We make extensive use of vast, high-speed networks that are ubiquitously available, at least in academic environments. And last, but certainly not least, hardware is cheap allowing us to build smarter devices on ever larger scales.

Although over the past decade we witnessed an unimaginable rate of growth in the technology sector, we have not changed our daily habits nearly to the same degree to take advantage of the new opportunities. Most of us are still not at ease with investing time, money and other valuable resources in the wide spread implementation of high-tech solutions.

Clever Laundry Room (henceforth, CLR) is a project that proves how straightforward it is to incorporate technology into the most basic and common aspects of daily life. It is our hope that it will inspire many other solutions to daily life issues and that it will convince the world at large of the benefits of ubiquitous presence of technology.

2. Overview

CLR is a solution for electronically monitoring the status of washer and dryer machines in laundry rooms across Duke University campus. The need for such a monitoring solution is prompted by the dynamic aspects of students' lives. There seems to be a pattern for doing one's laundry such that minima and maxima in work loads are very far apart. When the laundry room is busy it makes for an interesting balancing act as one forgets to move one's clothes from the washer to the dryer for instance. The personal nature of the process makes it awkward to remove another's clothes from a machine, pile them up in a potentially dirty spot so that one's clothes can now be washed or dried.

CLR will enable students to check from their dorm computers, on the web, if there are any available washers or dryers before carrying a huge laundry bag over just to realize they have to wait around for a machine to be done so that they can wait some more till the owner gets there to remove his or her clothes. CLR will also enable students to subscribe to notifications which will alert them either when a washer is available (in the scenario that all are busy) or when their washer or dryer is about to be done. This should make for faster turnaround during peak times.

Through the implementation of CLR we hope to attain several goals:

- (1) A solution to current issues faced by students living on campus when dealing with the common routine of doing laundry.
- (2) A technology showcase in support of web services and how they can be combined with hardware devices to allow (remote) access to smart devices as well as aggregation of generated data.
- (3) A technology showcase for a monitoring solution of equipment that is beyond the traditional realm of the IT industry.

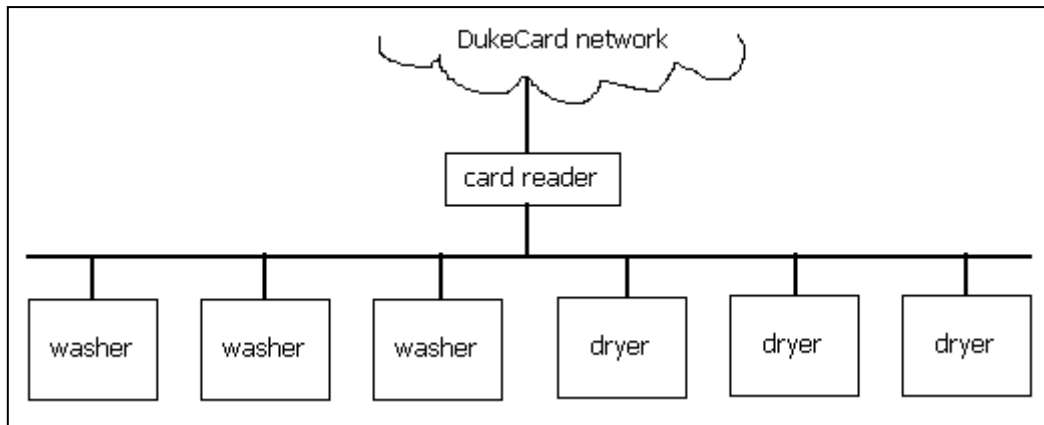


Figure 1. Existing laundry room setup.

3. Current setup

In our endeavor for building CLR we don't start entirely from scratch. Currently, the laundry rooms are integrated in Duke's electronic card payment mechanism. This allows a student to pay for laundry services using his or her DukeCard. The card reader is able to distinguish between available and busy machines and is able to instruct a machine to start upon debiting a student's FLEX account. This setup can be easily seen in Figure 1.

4. CLR Architecture

CLR will be composed from a hardware device and several software agents. The task of the hardware device is to aggregate the status of the washers and dryers such that external systems can query for this data. The software agents will interact with one another at various levels to allow active or passive monitoring of the machines.

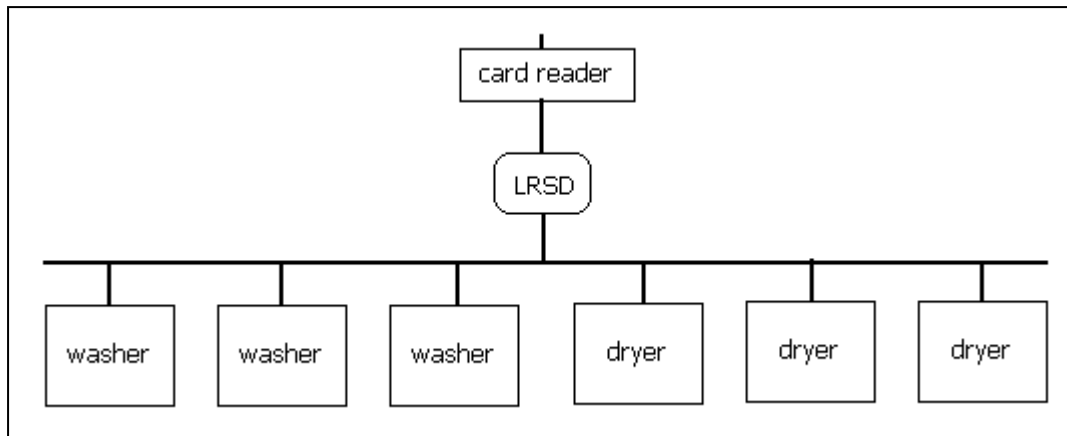
4.1. Laundry Room Smart Device (LRSD)

LRSD is a piece of hardware whose role, on one hand, is to interface with the washers and dryers in the laundry room. On the other hand, the device will be connected to an IP network so that it may take requests from and communicate the data to aggregation servers.

LRSD will be a CoolRoad Runner-II embedded PC (Lippert, 2002) based on the PC/104 Embedded-PC Standard (PC/104 Consortium, 2002). This platform uses the widespread x86 type processors, which enables us to run an embedded version of the popular operating system Windows 2000 or XP (Microsoft, 2002). We choose an embedded version of the OS as opposed to the standard PC version because a small footprint for the software allows for a cheaper hardware configuration of the device.

By fitting all the software on to a common size CompactFlash card we should be able to buy the LRSD hardware for approximately \$100 for each device. This includes the processor, memory and network card. There will be one device per laundry room. Details can be seen in Figure 2.

It is important to note that the LRSD does not affect in any way the current communication link between the washers and the card reader. It merely acts as a pass-through device from the perspective of both card reader and machine.



4.2. Software agents

The LRSD will run Microsoft's .Net Framework on top of the operating system. We chose this platform because it facilitates fast and easy development of web services. We chose web services because they provide for a standards-based interoperable mechanism for exchanging data between networked devices.

A first agent is the Laundry Room Agent (LRA) which will run on the LRSD. This is the piece of software that will communicate through the serial link with each individual washer and dryer to obtain their status. In turn it will be exposed as a web service through the IP-based network so that servers may query it for the collected data.

Next comes the Laundry Service Agent (LSA). This second tier of software has the responsibility of aggregating all data collected by the various LRAs around campus. In a similar fashion, it will expose the entire collection of data to the outside world using web services build on top of the .NET Framework.

The last tier of software will be the actual applications that will take requests from users and will ask the LSA to provide the relevant information. Since we expose the LSA as a web service, client applications can run on any platform and on any device. They can be web sites running on Windows 2000 or Solaris, they can be PDAs like PocketPCs and PalmPilots, or mere desktop applications. The entire architecture is presented in Figure 3.

The main venue for consuming CLR services will be MyDuke.com (Negritoiu, 2001). This website will provide a section for students to check the status and availability of washers and dryers around campus. It will also interface with Microsoft's .Net Alerts system (Microsoft, 2001) to provide notifications to users through tools like MSN Messenger. We chose .NET Alerts as the platform for the notification part of the project because there exists no other integrated development platform that provides such a service at this time.

Although in its initial stage the CLR project will be integrated with MyDuke.com, the web services provided by the LSA can be consumed from any application written by any other developer. The degree of openness will be decided at a later time based on input from all groups involved in the project.

5. Alternative Architecture

Another possibility for implementing CLR would be to extend the software running on the card readers to expose to authorized external clients the information it already collects. In this

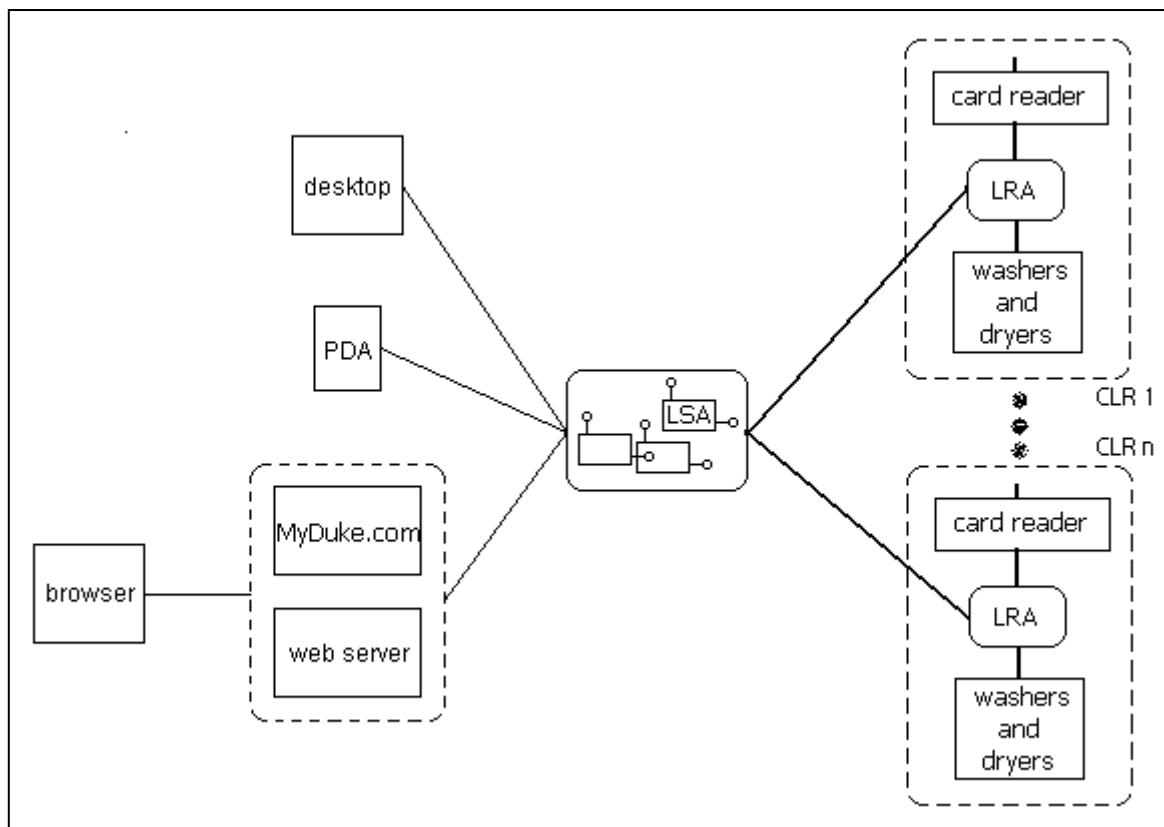


Figure 3. Architecture of CLR project.

scenario the LRSD would disappear and the LRA would become the extension to the existing software running on the card reader.

Although intuitively cheaper and faster to implement, this alternative architecture has the drawback of intervening with software changes on several devices connected to the DukeCard network. Due to security concerns and stability of a mission critical system we believe it's wise to stay away from this strategy. It is also difficult to estimate how much time software and potential hardware changes will take.

We believe that building a separate device like the LRSD to provide the data would ultimately result in a more secure and faster to implement solution. We also believe it will be easier to win the approval of the administration with the architecture presented in (4) because we do not affect in any way the current system.

6. Conclusion

CLR is a project that distances itself from traditional uses of technology. We hope to provide a useful service to the community and at the same time show how one can integrate such distinct devices to ultimately make life around us easier and less stressful.

References – Clever Laundry Room

1. Lippert Company, *CoolRoad Runner-II*, http://www.lippert-at.com/tools/pc104p/ds_pc104p_crr2.html
2. PC/104 Consortium, *PC/104 Technology Overview*, <http://www.pc104.org/technology/>
3. Microsoft, *Windows Embedded Family*, <http://www.microsoft.com/windows/embedded/default.asp>
4. Stefan Negritoiu, *MyDuke.com: A Technical Overview*, November 2001
5. Microsoft, *.NET Alerts*