IST-Africa

IST-Africa 2011 Conference Proceedings
Paul Cunningham and Miriam Cunningham (Eds)
IIMC International Information Management Corporation, 2011
ISBN: 978-1-905824-24-3

Beachcomber: Linking the "Internet of Things" to the "Internet of People"

Laurie BUTGEREIT, Dr Louis COETZEE
Meraka Institute, CSIR, Pretoria, South Africa
Tel: +27-12-841-3200, Fax: +27-12-841-4720
Email: LButgereit@meraka.org.za, LCoetze1@csir.co.za

Abstract: As more and more "smart things" get connected to the Internet, the "Internet of Things" grows. It is currently quite common that aircraft have transponders and cars have tracking devices. Even the family pet, FIDO, often has a chip embedded. Many of these "smart things" can also be remotely controlled via their Internet connection. Geysers (hot water heaters) can be turned off by power companies when under peak load. Cars can be immobilised remotely by security companies when they determine that the car has been stolen. Beachcomber is a platform designed to allow easy bearer agnostic communication between "things" and humans. Beachcomber enables "things" to communicate via a number of protocols (such as XMPP, HTTP, QR Code, etc) and enables humans to communicate via a number of other protocols (such as Email, XMPP, Twitter, etc). Beachcomber links "things" with their human owners in a bearer agnostic manner.
Keywords: IoT, "Internet of Things"

1 Introduction – Background on the Internet of "Things"

Human beings use "things". Scientists often maintain that one of the basic differences between humans and other members of the animal kingdom is that humans use "things" or tools and animals do not. Perhaps it is better argued that humans understand the casual relationship between "things" or tools and the result of their use [1].

Humans use "things" such as knives and forks for one of their most basic needs - eating. Most human sports are played with "things" such as balls, bats, clubs, tees, rackets, and nets. Even sports which initially appear to not use "things" still do. Consider boxing where the participants use boxing gloves. Even in one of the most elementary sports such as running, humans use specialised "things" such as special shoes and starting blocks. In modern society, even birth and death is often accompanied by the use of "things" or tools.

But although one could argue that some other members of the animal kingdom use "things" (such as higher primates), humans also exert a sense of ownership over their "things". Very few "things" are considered unowned. Humans own clothes, houses, and cars. The municipality owns the roads, road signs, health clinics, and offices. Depending on the type of government in place, the state (or province) may own other offices, larger hospitals, and schools. The national government may own military equipment and the unoccupied land.

Once humans own something, they often write labels on them and paint them in personal (or national, or corporate) colours. Even the copyright on this paper is a type of labelling. Eventually humans have too many "things" to keep track of and they started putting some sort of label on these "things" which computers could read.

Large supermarkets label all their "things" with bar codes which check out registers can read. Aircraft have transponders installed so that air traffic controllers can easily track those flying "things". Luxury motor cars have satellite tracking devices so that the owner or a security company know where those "things" are. Even FIDO, the faithful family pet dog, has a chip embedded so that the a friendly vet or dog shelter could find FIDO's owners in case FIDO wandered off.

Eventually humans have started controlling their "things" remotely. The local electricity company often has devices installed on geysers (or hot water heaters) to allow them to turn the temperature of the water down during peak load periods. Sophisticated traffic monitoring and control systems can remotely control robots (traffic lights) to enhance the flow of traffic or allow emergency vehicles speedy travel. Motor vehicles often have immobilisers installed to prevent car theft – and, if by any chance the car was stolen, the immobiliser can be remotely enabled to stop the car. The military has unmanned drone aircraft which are remotely flown by pilots sitting on the ground.

As more and more "things" become traceable and controllable remotely, the "Internet of Things" grows.

2 What is the "Internet of Things"

The term "ubiquitous" has become, well, ubiquitous, when talking about the Internet. Discussions about the Internet are peppered with phrases such as "ubiquitous computing" and "the ubiquitous cell phone". These phrases describe a situation where people are getting access to the Internet away from their traditional desktop computer configurations.

The "Internet of Things" could be considered to be the next step beyond "ubiquitous computing" [2]. The "Internet of Things" describes a situation where more and more "things" have embedded intelligence. These "things" will be considered to be "smart things". These "things" will be able to communicate with the Internet, thus allowing for their integration in Information Technology (IT) processes. These "things" will be able to communicate through a variety of methods including radio frequency identification (RFID) tagging, traditional Internet access, and use of bar codes.

The Internet of Things describes a vision where the lines between the real world and the digital world are blurred, where computing melts invisibly into the fabric of our business, personal and social environments, improving quality in our economic, health, community and private lives.

3 Beachcomber Research Project

The Beachcomber project was started by the Internet of Things Engineering Group at the Meraka Institute, CSIR, South Africa. The Beachcomber project's goal was to create an architecture and platform which would enable easy communication between this new "Internet of Things" and the people who owned the "things". The research question was:

How can Beachcomber be architected and implemented to enable bearer agnostic communication between the "things" and their owners via an intermediary business service?

This research question could be divided into four sub-questions:

1. How can Beachcomber be architected to enable bearer agnostic communication with "things"?
2. How can Beachcomber be architected to enable bearer agnostic communication with the owners of these "things"?
3. How can Beachcomber be architected to enable bearer agnostic communication with various business services which communicate with "things" and people?
4. How can these three architectures be combined?

By answering these four sub-questions, the primary research question would be answered. The research objective was b:

> *To deliver an architecture and implementation which allows bearer agnostic communication between "things", their owners and intermediary business services.*

This objective could also be divided into four sub-objectives:

1. To deliver components which enable bearer agnostic communication with "things"
2. To deliver components which enable bearer agnostic communication with people
3. To deliver components which enable bearer agnostic communication with business services which communicate with both people and "things"
4. To link these three sets of components together.

By satisfying the four sub-objectives, the primary objective would be satisfied.

4 Design Science

One of the expected outcomes of research in Information Systems is to produce knowledge that enables the application of Information Technology to solve managerial problems and organisational problems [3]. There are two complementary research paradigms in Information Systems research: Behavioural Science and Design Science. Behavioural Science is based in natural science and attempts to find the *truth* about something. Design Science is based in engineering and has as its goal the *utility* of something.

Design Science research (as opposed to Behavioural Science research) was chosen for the Beachcomber project because of this difference. The Beachcomber architecture and implementation needed to be a useful *utility*.

Design Science research has produced four different types of artifacts: Constructs, Models, Methods, and Instantiations. Constructs provide the language in which problems and solutions are defined and communicated. Models use constructs to represent real-life situations. Methods define solution processing and algorithms. Instantiations are implementations of constructs, models, and methods.

The Beachcomber project produced a model and an instantiation of that model. The model is an architecture of how best to provide bearer agnostic communication between "things", people and intermediary business services. The instantiation is an actual implementation of that model.

Hevner [4] describes seven guidelines in Design Science research:

1. A viable artifact (construct, model, method, or instantiation) must be created.
2. The artifact must solve an important problem.
3. The artifact must be rigorously tested.
4. The research must create clear and verifiable contributions.
5. Rigorous research methods must be used in the construction of the artifact.
6. The domain must be searched for solutions to the problem.
7. The research must be reported to technical and non-technical audiences.

The Beachcomber project ensured that these seven guidelines were adhered to. By adhering to these guidelines, Beachcomber can be classified as an output of Information Technology *research* and not just Information Technology *development*.

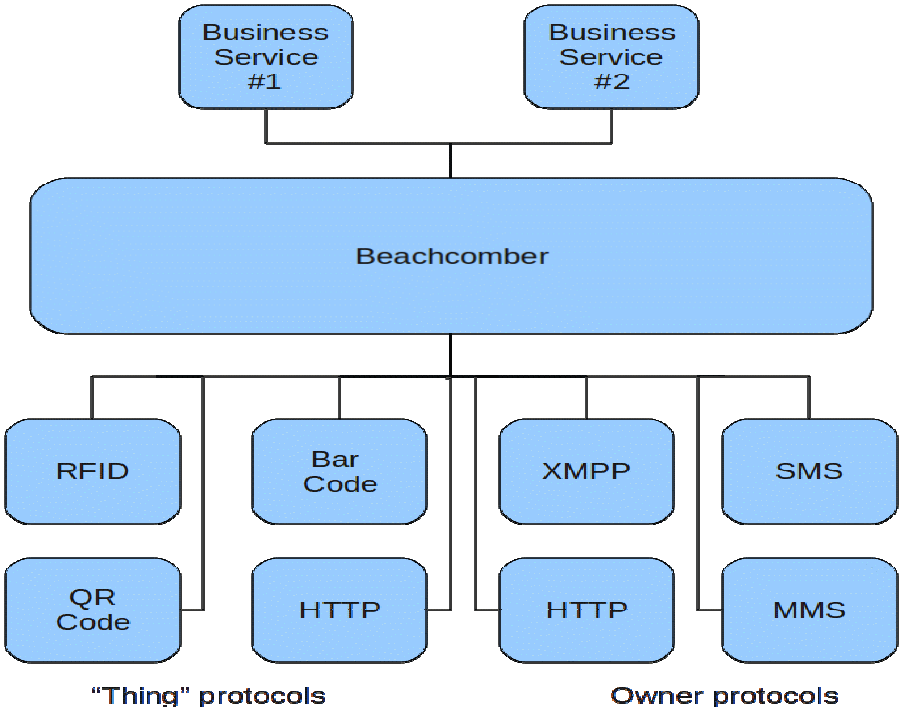
5 Beachcomber Model Artifact

In providing for bearer agnostic communication between "things", the owner of these "things", and an intermediary business service, Beachcomber needed to be able to accept information using a wide variety of communication protocols. In other words, some "things" may be able to communicate using RFID tags or by using commercial bar codes. Other "things" may be able to communicate using Internet protocols or by using QR (Quick

Response) codes. Beachcomber needed to be able to accept all of these communication protocols and convert them into a suitable format.

In addition, the owners of these "things" may wish to communicate using a wide variety of protocols. For example, some owners may wish to communicate using XMPP (eXtensible Messaging and Presence Protocol) or Jabber (Instant messaging). Other owners may wish to communicate using SMS (Short Messaging System) or MMS (Multimedia Messaging Service). And still other owners may wish to communicate using HTTP (HyperText Transfer Protocol) protocols. Beachcomber needed to be able to accept all of these communication protocols and convert them into a suitable format.

The newly formatted message would then be forwarded onto which ever business service needed to keep track of these "things" or the owners of these "things". Drawing 1 shows the basic design of Beachcomber.



Drawing 1: Beachcomber model

The "Internet of Things" involves high numbers of things. One can grasp the idea of quantities by imagining the number of consumer products manufactured and sold every year. One interesting calculation done by Prof Elgar Fleish of the University of St Gallen in Zurich was to take the revenue of a leading consumer goods company and divide it by the estimated product price. That number was then expanded to an estimated 100% market saturation. The resulting number (84 billion) only begins to hint at the dimensions of the "Internet of Things" [5].

For that reason, wherever possible communication between the components would be asynchronous and decoupled. Asynchronous and decoupled communication would support scalability [6]. Scalability would be a crucial feature of Beachcomber. However, all blocks in the diagram shown in Drawing 1 also support synchronous communication in addition to asynchronous.

6 Beachcomber Instantiation Artifact

The initial implementation of Beachcomber was designed and written as a JEE (Java Enterprise Edition) application using JBoss as an execution container and hosted on a Linux operating system. Both Linux and JEE/JBoss are horizontally and vertically scalable. Linux has been hosted on numerous hardware configurations from embedded robotic arms [7] to large superclusters [8]. JEE was designed for scalable, reliable and secure enterprise applications [9]. JBoss is an open source implementation of JEE [10, 11]. Mobicents was used for the actual framework within JBoss. Specific protocols were implemented as Resource Adaptors (RAs). Business services were implemented as Service Building Blocks (SBBs).

The common Beachcomber message format as described in section 5 contains the following information:

- *fromAddress* - The address of the device which was sending the information
- *toAddress* – The address of where the information is being sent
- *message* – The actual textual message
- *filename* – In the case of images, sounds, and other multimedia messages, the name of the file holding the contents
- *contentType* – In the case of images, sounds and other multimedia messages, the type of the content being held in the file

For example, a "thing" would have some unique address which would be encoded into the *fromAddress* field. This "thing" would then send this information to Beachcomber via some specific channel which would be identified by the *toAddress* field. The field m*essage* would contain any textual message which would be transmitted while *filename* would contain the name of the operating system file containing any message portion which was not textual – for example, an image or a sound clip. The type of information stored in the file would be indicated in the *contentType* field.

The "thing" information (in a format specific to the individual "thing") could be sent either asynchronously (preferred) or synchronously (when only absolutely necessary) from the "thing", through some specific translation block which would convert the information into a common Beachcomber message and forward it to Beachcomber. Beachcomber would inspect this message to decide which logical business service must receive this information and forward it appropriately. The logical business service would process the information and reply in a common Beachcomber message format. That reply, if necessary, could be forwarded back to the protocol translation block and then sent back to the specific "thing".

The initial implementation of Beachcomber supported two "thing" protocols: HTTP and QR Code. These protocols allowed things to communication with Beachcomber. The initial implementation of Beachcomber also supported five human protocols: POP3 Email, XMPP, HTTP, Twitter and MXit (MXit is a mobile chat based protocol similar to XMPP but optimised for a mobile environment where bandwidth is at a premium). These protocols allowed humans to communicate with Beachcomber.

The initial implementation of Beachcomber supported a handful of independent business services.

The business service Tony, named after St Anthony, the patron saint of lost "things", attempts to allow owners to find their "things" which have been physically lost. The people communicated with Tony through a variety of protocols including email, XMPP, and MXit. The "things" communicated with Tony through QR codes. When the QR codes where photographed through the use of a smart phone, GPS coordinates were also sent with the URL which was encoded in the QR code to give the physical location of the "thing" that was lost and found.

The business service GreenIT keeps track of electrical power usage, core temperatures, and battery levels of various computers within the organisation or under control of the organisation but at remote locations. Computer users could easily participate in the GreenIT project by installing a small script on their computer which would communicate periodically with the Beachcomber platform about the electrical consumption and current temperatures.

The business service RSAWORKS is an attempt to get a large number of "things" in South Africa to post their status on Twitter. This is an ongoing publicly available project and the various statuses can be viewed by following the Twitter user "rsaworks". This project includes a number of privately owned houses which tweet their electrical consumption and a number of privately owned weather stations which tweet weather conditions.

The business service Defoe, named after Daniel Defoe who is often attributed to being the first modern journalist, enables community journalists to compile short articles about community events and information and submit them to the Beachcomber platform using a variety of protocols. Defoe maintains a magazine of these short articles and prints out QR Codes which can be distributed on bulletin boards around the community to enable the public to easily read the magazine.

7 Design Science Evaluation

The seven guidelines for Design Science research have been satisfied with the Beachcomber project:

1. Beachcomber is a viable artifact in terms of Design Science. Beachcomber is a model of how to design a bearer agnostic platform to enable communication between "things" and their owners. It is also an instantiation of that model.
2. Beachcomber solves an important problem of how people can keep track of "things" in a bearer agnostic manner.
3. Beachcomber has been rigorously tested with small pilot programs utilising a number specific business services hosted on Beachcomber.
4. Beachcomber has made a clear contribution in the area of bearer agnostic communication between "things", people, and intermediary business services by translating message formats for a specific protocol (such as HTTP) into a common message format which could be understood by all business services hosted on Beachcomber.
5. Rigorous methods were used in the construction of Beachcomber.
6. The "Internet of Things" domain, along with more traditional Information Technology domains, were searched for best solutions to problems encountered during the construction of Beachcomber.
7. The best practices and lessons learned on Beachcomber will be presented at IST-Africa to both technical and non-technical audiences.

Beachcomber satisfies these requirements and can be classified as an artifact in terms of Design Science.

8 Conclusion

Beachcomber is a platform which supports bearer agnostic communication between "things" and their owners. Multiple types of protocols can be used for communicating with "things" such as QR Codes, bar coding, and HTTP. In addition, multiple types of protocols can be used to communicate with the owners of these "things" such as XMPP, Twitter, and Email. Beachcomber converts all these protocols into a common message format and forwards them on to a business service for processing. The business service can then reply

and, if necessary, that reply will be translated into the required format for forwarding to a specific “thing”.

This creates a customisable platform allowing pluggable components to be added supporting a wide variety of protocols. This makes it easy to link the “Internet of Things” to the Internet of people.

This is an ongoing research project. To date five protocols are supported by the Beachcomber platform: HTTP (both client and server including QR Codes), POP3 Email, XMPP, Mxit, and Twitter. In addition, nine business services were implemented on the platform. These business services include monitoring home electrical consumption, monitoring information at personal weather stations, and assisting in finding lost “things” which owners have properly identified.

References

[1] R. Peeters, L. Simone, K. Nelissen, M. Fabbri-Destro, W. Vanduffel, G. Rizzolatti and G. A. Orban. (2009, The representation of tool use in humans and monkeys: Common and uniquely human features. *Journal of Neuroscience 29(37),* pp. 11523.

[2] ITU, "The internet of things, executive summary," International Telecommunications Union, Geneva, 2005.

[3] A. R. Hevner and S. T. March. (2003, The information systems research cycle. *Computer 36(11),* pp. 111-113.

[4] A. R. Hevner, S. T. March, J. Park and S. Ram. (2004, Design science in information systems research. *Management Information Systems Quarterly 28(1),* pp. 75-106.

[5] E. Fleisch. (2010, What is the internet of things? *When Things Add Value.Auto-ID Labs White Paper WP-BIZAPP-053, Auto-ID Lab St.Gallen, Switzerland* .

[6] M. Abbott and M. Fisher, *The Art of Scalability.* Boston: Addison-Wesley, 2009.

[7] L. Torvalds. (1999, The linux edge. *Commun ACM 42(4),* pp. 38-39.

[8] R. Bryant and J. Hawkes. Linux® scalability for large NUMA systems. Presented at Linux Symposium.

[9] S. Kounev. J2EE performance and scalability-from measuring to predicting. Presented at Spec Benchmark Workshop.

[10] M. Fleury and F. Reverbel. (2003, The JBoss extensible server. *Lecture Notes in Computer Science 2672/2003*pp. 344-373.

[11] N. Richards and S. Griffith, *JBoss A Developer's Notebook.* Sebastopol, California: O'Reilly, 2005.