Ambient Earth

Internship report

Christian Luijten
christian06@ru.is
c.a.a.m.luijten@student.tue.nl

11 November 2006

CADIA group
School of Science and Engineering
Reykjavík University, Reykjavík

Under supervision of:
Kristinn R. Thorisson, Ph.D. (Reykjavík University)
dr.ir. Huub van de Wetering (Eindhoven University of Technology)

Christian Luijten (christian06@ru.is)

Preface & Acknowledgements

During the period of September to November 2006 I have been working at the Center for Analysis & Design of Intelligent Agents (CADIA) research lab of the School of Science and Engineering of Reykjavík University (RU) in Iceland.

I would like to thank Kristinn Thorisson of RU for giving me this opportunity and the useful input during the discussions we had before and during the project.

The internship was supervised from the Technical University Eindhoven (TU/e) by Huub van de Wetering, who gave me useful hints and tips throughout the design and documentation phase and I would like to thank for that.

When I considered doing part of my studies abroad, there was no agreeement with any Icelandic institution yet. I had thought of an internship at a Scandinavian university and by coincidence I discovered that - just weeks before - an agreement was signed between TU/e and RU.

From this position I would therefore also like to thank Jan-Friso Groote (TU/e) and Luca Aceto (RU) for initiating this agreement.

Others who made my stay more pleasant were the people from the International Offices of both RU and the University of Iceland, my fellow students both at CADIA and the Erasmus Intensive Language Course in August and everybody else I met during this period.

I have never experienced anything like this before. It was a unique experience which I was happy to share with you all.

Thank you all very much! Takk fyrir!

Reykjavík, November 2006
Christian Luijten

Contents

1 Introduction

This document describes the activities concerning my internship at Reykjavík University. The introduction, which you are reading right now, describes the structure of this document.

Chapter 2 describes the initial assignment as given by Kristinn R. Thorisson of Reykjavík University. Since the final product differs from this first idea, the changes and their reasoning are given.

In Chapter 3 a planning is proposed. It is taken from the planning as written in the logbook which I held online at `http://nemendur.ru.is/christian06/logbook.html`.

The design of the system to be built is given in Chapters 4, 5, and 6. These chapters are included from the design document. The full document can be downloaded from `http://nemendur.ru.is/christian06/amber/design.pdf` or the most recent version can be requested directly from the author, preferably by E-mail.

The complete internship is evaluated in Chapter 8. This will contain learning points, which aspects went according to expectations and what can be improved.

The document itself is concluded in Chapter 9.

2 Assignment

The original assignment as written up by Kristinn Thorisson:

> The "Semantic Ambient Web Monitor" project's goal is to give people a sense of online acitivity and community interest regarding particular topics of interest and discussion. The prototype will have a simple small area where elements related to online activity – each dot representing e.g. a posting on an online forum or a blog site about the particular topic of interest. Each corner represent a concept related to the topic; If the topic to be monitored is "artificial intelligence", one corner might represent "humanoid robots", another might represent "artificial neural networks", a third could be "expert systems" and a fourth could be "machine learning". As the dots move they get pulled stronger towards the center the older they are; the more they relate to a concept the closer they get pulled towards the corner representing that concept.
>
> This produces very "holistic" patterns that can be interpreted in an instant by an onlooker: An equal distribution of dots would mean that all concepts are discussed equally, on average, in all postings. An oval orbit would mean that the majority of postings are about the concepts in oposite corners of the square, etc. If the dots left trails one could see how things change over time, e.g. days or months. Snapshots of the square, e.g. one per day or month, would give you a visual history of how the disucssion has changed over time regarding those concepts.
>
> The methodology that will be used to build the monitor enables a very incremental, modular approach. The software will be made open-source and thus others can build monitors for other topics than those chosen in this project.

While the assignment text hasn't changed during the project, some aspects of it were implemented differently.

- Stories aren't pulled back into the center the older they are because it serves no visualizing purpose and only clutters the image. At the end of their lifetime, particles quickly fall back to the center.
- Dots are indeed leaving trails. Not to show changes in time, but to create a quieter image which is easier to understand and to follow.

3 Planning

The actual internship period amounts to 14 weeks and starts week 34. Before the actual working period has started, the assignment was already largely agreed upon.

A week by week planning and logbook was held during the internship at the web address http://nemendur.ru.is/christian06/logbook.html.

3.1 Weeks 34 and 35: Introduction

Plan Learning the organization, people, methodologies and procedures.

Actions According to plan, read and talked a lot. Also did experiments with Psyclone.

3.2 Weeks 36 and 37: Global architecture design

Plan Having a global design of the modules and a first bit of object model, meanwhile playing around with posting and retrieving messages with Psyclone.

Actions Mainly work on design, also created first version of the Psyclone specification file. Setting up project in Eclipse, preparing codebase. Laptop got problems, installed a desktop computer. First simple prototype of visualizer module (single dot orbiting a circle).

3.3 Weeks 38—44: Detailed design and coding

Plan Create detailed design and code, starting with basic visualization, then the crawler module, the analysis module and then the final visualizer. Each module should at most take two weeks to implement.

Actions Fully implemented all modules, leaving only a few minor features which were implemented in week 45 during refactoring.

3.4 Week 45: Spare week for coding

Plan Refactoring, fixing bugs, completing Javadoc code documentation.

Actions Preparing work for internship report document and source code documentation for the design document.

3.5 Weeks 46 and 47: Finishing up

Plan Fine-tune crawling, analysis and visualization parameters. Create final distribution files. Test those on various platforms. No more large design and source code changes; only bugfixing.

Actions

3.6 Week 48: Back in Eindhoven

Plan Finish internship report and paperworks.

Actions

4 Usage scenarios

4.1 A story is posted to a weblog

When a story is posted on a weblog, it will show up in its RSS feed (this happens of course outside of our responsibility). If AMBER is monitoring this particular weblog, it will retrieve the story and analyze it. The story is then displayed on a screen using the results of the analysis.

To get a more concrete idea, suppose AMBER is monitoring various A.I. related weblogs and we would like to find out what they are mainly writing about. We configure the analysis component in such a way that it can decide whether a certain subject is dealt with in a story, thereby creating a profile for every story. Stories with similar subjects will then show up close to eachother in the display, stories with orthogonal subjects will be very far apart.

The result is an image with various "clouds" of in some way related stories.

4.2 A discussion is held on a web forum

Discussions on web forums can get lengthy and the main subject can change multiple times during their lives. To get an idea what subjects the whole discussion has been about, AMBER can show a cloud map of (part of) the discussion. It could even show an animation to show how the discussion developed over time.

Using the animated view of the discussion development, a new participant in the discussion can decide upon whether bringing up an old discussion point is a good idea or not. It is also a way to locate a certain subject within a long list of replies.

5 Requirements

There are various kinds of requirements to be identified. A distinction can be made between functional and extra-functional (or non-functional) requirements.

5.1 Extra-functional requirements

1. The system must make use of the Psyclone framework for communication.
2. The system will be implemented in the Java programming language.
3. The display module with the Java Applet must be able to run on any machine with a properly installed and recent Java Virtual Machine (i.e. not only on the machine running the rest of the system).
4. It must be possible to add modules with similar functionality to operate in parallel with modules already there. For example when the Java Applet is running, it should also be possible to have the full screen module running at the same time.

5.2 Functional requirements

These requirements describe which *inputs*, *outputs*, *storage* and *computations* exist in the system and how they are *timed and synchronized*. Finally, since this is a very important part of the project, there are two separate sections on *story analysis* and *visualization* requirements.

5.2.1 Inputs

1. The system must be configurable to specify which sources will be monitored.
2. The system will use the configuration to get information from the internet from the specified sources.
3. Configuration of the system goes via Psyclone using module parameters.
4. Parts of the system must accept triggers from Psyclone whiteboards.
5. Sources must be RDF Site Summary (RSS) feeds, possibly aggregated via an Outline Processor Markup Language (OPML) file. The system should however be

prepared to support other source types as well (i.e. it should be easily expandable).

6. The Applet display is non-interactive (no input).

5.2.2 Outputs

1. There is an output module which is to be used within a website, i.e. a Java Applet.
2. There is an output module which runs standalone and in full screen and displays more information than the Applet can.

5.2.3 Storage

1. The system on itself does not store anything.

5.2.4 Computations

1. The system must decide of a delivered story what its subject(s) is/are.
2. The system may put weights on the subjects instead of a boolean value.

5.2.5 Timing and synchronization

Synchronization between modules is handled by Psyclone, so no requirements need to be added to the system itself.

5.2.6 Story analysis

1. When stories come in, they are analysed by analysis modules.
2. Every module adds some meta-information to the story depending on the module analysis.

5.2.7 Visualization

The following requirements are common for both the applet and the standalone viewer.

1. A story is represented as a dot.
2. In the center of the display is Earth (with picture?).
3. Dots are launched into orbit around Earth.

4. The dots leave fading trails as they move.
5. There are some small, heavy bodies in "geostationary" orbit around Earth representing values of an enumeration of meta-information (for instance story subjects). They attract the stories depending on how much they match the story's subject.

There will be two different views, a static and a dynamic one. Which one is used depends on the application. To get an idea of the activity at a certain moment in time, the static view is used. For a "real-time" view of internet activity, the dynamic view can be used.

The term "static" doesn't mean the image is standing still, it will behave exactly the same as the dynamic view. However, some physical laws don't apply or are differently calibrated in order to give a constant image. In other words, while in dynamic view stories can appear and disappear, in the static view the stories are a given constant.

Differences between Applet and Standalone viewer

1. The applet display will in practice be considerably smaller than the standalone viewer. Therefore, the applet is less detailed and some physical laws might need to be bend a bit.

6 Architecture

The architecture of AMBER is defined in terms of modules, whiteboards and the messages they use to communicate. Figure 6.1 shows the flow of messages between the modules and whiteboards.

In the Section 6.1 the modules are described and in Section 6.2 the messages connecting them are defined.

6.1 Modules

A complete AMBER system will comprise at least three modules running at the same time; there is a Crawler module, an analysis module called Sieve and a display called ShowOff. The modules are separate executables with their own life-cycles and resources. Since TCP/IP is used, the executables don't need to be on the same machine to communicate.

Every module has a specified interface through which communication with Psyclone is handled.

6.1.1 Crawler modules

When the Crawler is started, it will create one of the available handlers (depending on what is specified on the command line or what is set as default during build time).

It also creates an AirBrush instance to communicate with Psyclone via JavaOpenAIR. The module name announced to Psyclone is 'Crawler.' plus the name of the handler, so 'Crawler.RSS' in case of the RSS handler.

After connecting with Psyclone, the handler can get its parameters stored in the psySpec file and go to work. It will post stories with type 'Story' on the whiteboard 'WB.Stories'.

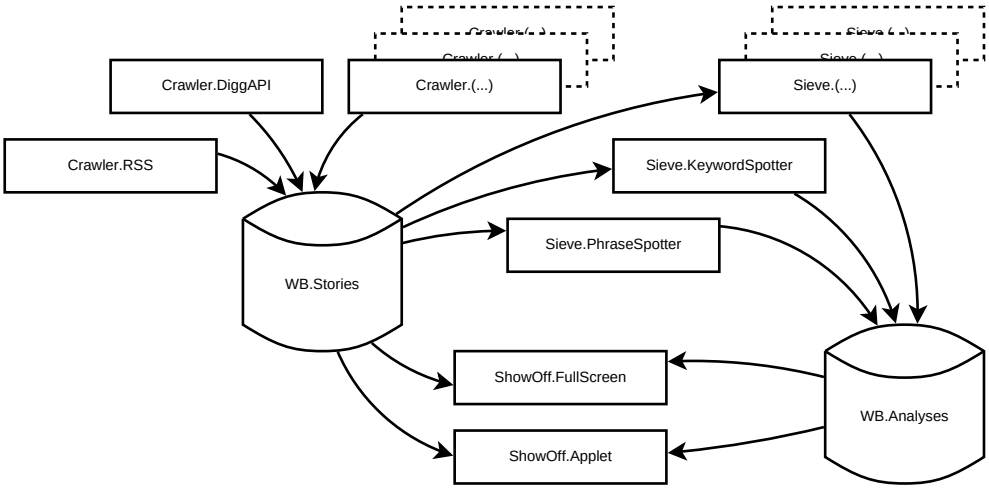

Figure 6.1: Global AMBER architecture, the names are Psyclone module names

RSS

The RSS crawler module will be fairly straightforward. It fetches the RSS feed from a set URL and produces a Story message for every new item that appears. The contents of the Story message is specified in Section 6.2.

Although the module is called RSS, it can handle Atom feeds, which is also quite a popular format.

Psyclone module specification for module Crawler.RSS:

Whiteboard	Type
Triggered by WB.Control	Feed.*
Post to WB.Stories	Story

DiggAPI

Digg is a website which lets users submit stories found on the web. Other users then moderate the submissions either by 'digging' or 'burying' a story. A story with a lot of 'diggs' is a popular one. The nice thing about Digg is that it actually does a lot of preprocessing work for the AMBER system.

Digg announced[1] that they will publish a public API within the next months. If time allows, a DiggAPI module is created.

[1] http://diggtheblog.blogspot.com/2006/07/digg-labs-launches-alpha.html

Psyclone module specification for module Crawler.DiggAPI:

Whiteboard	Type
Post to WB.Stories	Story

6.1.2 Sieve modules

All analysis modules, or sieves, will get a trigger from a new story on the whiteboard WB.Stories. They analyse it and if it can say anything about the story, an Analysis message is sent to the whiteboard WB.Analyses containing its judgement on the story.

The contents of this message is specified in Section 6.2.

Analysis modules may take any time they like to come to a verdict, but it is possible that a story has already disappeared from the visualization if the response is very late.

Since all modules regardless of their functionality employ the same external behaviour, the Psyclone specification is the same for every one of them.

Psyclone module specification for module Sieve.???:

Whiteboard	Type
Triggered by WB.Stories	Story
Post to WB.Analyses	Analysis

6.1.3 ShowOff modules

The ShowOff modules are visualizers which combine the crawled stories from the Crawler with the analyses from the Sieve modules.

Psyclone module specification for module ShowOff.???:

Whiteboard	Type
Triggered by WB.Stories	Story
Triggered by WB.Analyses	Analysis

Full screen

The full screen application shows the particles orbiting the earth, being attracted to various attractors hanging around it, depending on their topics.

In future versions it is possible to get extra information about stories by hovering over the particles (i.e. it should not be impossible to implement this within the current architecture).

Ambient applet

The ambient applet will display a very easy to understand image (a glance at it should be enough) of the status of the page it is on. I.e. if the page is a weblog, it should display subject information on that weblog, if it is on the page of a thread of a forum, it displays the flow of the discussion[2].

The applet has no high priority, and it would be enough to convert the full screen application to an applet.

6.2 Messages

There are a few message types in the system. Two of which must be defined system-wide because they are used in the communication between modules.

6.2.1 Message type 'Story'

The Story message is only posted to the whiteboard 'WB.Stories' and only by Crawler modules.

The message content is a YAML Ain't Markup Language (YAML)[3] document which represents the storyData field inside the Java counterpart of the message. It contains at least the properties 'URI' (to identify the story, GUIDs are RSS specific and cannot be used), 'Author', 'Title', 'Story-Content'.

It may also contain 'Publication-Date' (which is the date of publication in Internet Message Format[Res01, Section 3.3]), 'Kind' and other fields.

An example of a YAML document containing Story data:

[2] Because currently there is no way to track discussions due to lack of meta-information about how comments relate to eachother, this part won't be implemented

[3] http://www.yaml.org/

```
---
URI: http://ijsland.luijten.org/2006/09/12/skyr-wasdanou/
Author: Christian Luijten
Title: Skyr... Wasdanou?
Publication-Date: Tue, 12 Sep 2006 21:03:54 +0000
Kind: weblog-posting
Story-Content: >
  Een van die dingen die bij een onbekende cultuur horen zijn de
  eetgewoonten. Elk land heeft zo z'n producten die je nergens
  anders kan krijgen. IJslands nationale zuivelproduct heet Skyr,
  elke oma kan het maken, al is het nogal een hoop werk. Daarom is
  het lange tijd (lees: gedurende de jachtige periode na de tweede
  wereldoorlog toen de Amerikanen hier de boel kwamen ophaasten)
  in ongebruik geraakt, maar op een gegeven moment kwam de vraag
  toch weer terug en zijn een aantal zuivelproducenten het
  industrieel gaan produceren.
```

6.2.2 Message type 'Analysis'

Analysis typed messages are posted on the 'WB.Analyses' whiteboard only by Sieve modules. They contain information about stories present on the 'WB.Stories' whiteboard.

The content of these messages is also YAML format. Story messages are coupled with Analysis messages through their 'URI' fields, so this must be present.

An example of a message issued by an analysis module checking for the topic 'Zuivel' (which means dairy products in Dutch):

```
---
URI: http://ijsland.luijten.org/2006/09/12/skyr-wasdanou/
Topic: Zuivel
Relation-Strength: 1.0
Author-Strength: 0.1
```

Its 'Relation-Strength' suggests high relevance of the content with the topic. However, the 'Author-Strength' suggests that the author isn't an authority in the field.

Every analysis module sends a message to the whiteboard if it thinks it is relevant. It is thus possible that the same URI will get multiple analysis results or nothing at all, the visualizer module must cope with this and merge the available information.

7 Detailed design

The fully documented source code with detailed design can be found in the design document which is available at http://nemendur.ru.is/christian06/amber/design.pdf or directly from the author.

8 Evaluation

8.1 Learning points

The approach used in the development of this project is called Constructionists Design Methodology, which is a form of incremental design. For this project, this was a good choice, because many things were undecided upon in the beginning and only became clear after some first experiments.

One example would be the visualization of "replies to a story" which was initially thought of as adding a particle on a tail tied to the original story. The longer the tail, the more important a story would appear.

However, there is no connection between a reply and a story other than a humanly constructed one (there is no field in RSS which specifies it). It was thus impossible to create this link without going far beyond the scope of this project.

Instead, it was chosen to give all replies their own orbiting particles and if a story is popular, many particles appear and in this way visualize activity in a topic.

8.2 Points for improvement

It is very hard to create a realistic planning and to estimate one's capabilities right. Some details had to be simplified on the way to make the project fit within the available time, while other aspects which seemed complex were in fact solved very quickly.

9 Conclusion

During this internship, I created a simple framework for Internet discussion analysis and visualization. Along with the framework, I provide three modules that illustrate the possibilities within the framework.

Both modules and framework are extensible or replacable in various ways. The current visualization has a strong emphasis on stories as an entity. To get a visualization of all activity on the internet, a sample should be taken and the orbiting particles will no longer be entities on their own. They rather turn into classes of entities, which probably calls more for a cloud-like visualization.

One of the ideas is to change Ambient Earth into Ambient Sun, where the Sun's corona (the brightly shining part which is visible during a total solar eclipse) is taking over the role of the particles orbiting Earth.

9.1 Unimplemented parts

It is not yet possible to "travel through the past" as was an initial idea. This feature would enable a user to set a certain time in the past and see the situation at that point. A very simple approach would be to create screenshots of the display for every day and then display the desired one.

9.2 Points for expansion

The nice thing about using a message and whiteboarding system like Psyclone is that every component can be replaced fairly easily.

Currently, there is only one analysis module which can not make very intelligent decisions. The fact that this is a project within the AI department, makes it likely that someone will implement a smarter analysis module which can run in place or alongside the current one.

It is only possible to get stories from RSS feeds, but another module could be created which gets information out of newsgroups. A lot of AI discussion is going on on Usenet, so this might also be an interesting expansion of the system.

If the applet is to be used in a forum, a module with direct (read-only) database access could be created.

The applet could also be used in software development, displaying for instance build status, bugreports assigned to programmers, server load etc.

Then, finally, the visualization itself can be replaced by another one and one of the ideas is to make it look more like the Sun's corona, expanding it in those directions where much discussion is going on, shrinking it where discussion is quiet.

Bibliography

[Res01] P. Resnick. *RFC 2822 - Internet Message Format*, 2001. ftp://ftp.rfc-editor.org/in-notes/rfc2822.txt.