XO Lab Precursor Essay

Matt Adair, s3204584

Cost expressions small and,

reactions modifying to easy.

Introduction

My honours research relates to the intersection of human and machine, specifically the role of privacy in an environment of pervasive computing that utilises wireless sensor networks and neural-net learning algorithms that are intended to classify, predict and encourage types of human behaviour. To investigate this I will be examining mobile phones and their interactions with Internet-of-Things (IoT) devices that emit near ultra-high frequency audio signals. These signals invoke computational processes and network connections for the purpose of allowing communication between machine and machine, without user knowledge or interaction.

As a member of XO Lab I have been encouraged to investigate some of the more abstract ideas around the concept of human and machine interaction, including questions of human agency, identity and privacy. The way these questions are framed and analysed is vital for any attempt to navigate path through a complex field of study. I created an object through programming that could be explored and analysed by myself and the other members of the lab. This object (see Appendix A) took the form of some modified server-side code that was intended to demonstrate what human readable code for machine execution looks like. As an aside it should be noted here that on multiple occasions throughout

this Precursor research I have deliberately performed an action that could be considered an act of human agency.

As well as this object, created in the php webserver language, I experimented with Martin Niemöller's famous quote translated into other machine mediated languages such as hexadecimal, base 64 encode, morse code, SMS alphabet and binary. This quote which I read as a declaration of human identity in oppposition to classification, was particularly relevant to the conceptual framework of my research.

This exercise in translating text into various machine forms seemed too simplistic and was not particularly imlluminating. In response to suggestions from lab members, who noted that some of the code appeared as poetry and that code is an example of human agency, I refined the focus of my investigation. To do this I decided that some form of machine generated poetry could be useful in helping to determine how to *make strange* the human-machine interaction research I am interested in.

Precursor 1

In order to create a program that writes poetry it required codifying what a poem is, or could be. It also required a methodology for the program to procedurally follow in order to generate a poem. As my research is in the area of internet connected machines interacting with humans I chose to research this poetry writing problem on the internet. After finding a useful website on the construction of poetry and stanzas, another website was found that was devoted to word lists for scrabble enthusiasts. These

word lists were parsed into a series of files that would form the read only memory of the poetry writing web bot.

Given the unwieldy vocabulary of 97,648 words, I determined that a "random method seemed to be better than the systematic" (Turing 459) for the construction of the poem. The code (see Appendix B) was written to function on a web page that would load and run the code as soon as the page was rendered in the browser. Several additional functions (see Appendix C) were added throughout the versioning process which are listed as follows:

- additional vocabulary source based on Turing's paper and a Facebook contagion experiment paper (Kramer 8788)
- conversion of the poem to an alphabet of near ultra-high frequency values from 18000 Hz to 19875 Hz.
- HTML 5 audio synthesis code to transmit the poem
- send poem form the webserver via Google Cloud Messaging to a mobile phone

To demonstrate this precursor, I devised a performance during which I manually typed words into the audio synthesizer programmed to approximate the human voice. This formed a narrative of questions and statements that were intended to reflect my own uncertainty about the relevance of certain conceptual problems to my project. The comfort of being in control of the process, even as a performance, is bound up with problems of identity and human agency, prompting the question: would

the removal of this control be noticable in any state. For me, acting as the archetype in this instance, I needed to be in control, to ensure its subservience to my (human) will. In order to contextualise my response to within an array of other human responses, I needed to bypass or ignore reflective reasoning and assume the mantle of dispassionate observer. The rationale for this, as far as I am concerned, is that, ultimately, I cannot be the archetype.

Precursor 2

The lab feedback for the final version of Precursor 1 was to investigate a way of determining any meaning behind the poem created by the bot. To do so I utilised Natural Language Processing (NLP) as a method of deriving an ontology. Processing a poem via NLP required setting up a machine-learning program that used a lemmatizer, morphology tagger, a dependency parser and a precompiled 450MB English language model.

Running the program and examining the results demonstrated the complexity of NLP algorithms and results. It also revealed that this particular branch of investigation was an unproductive digression. The most useful outcome of this iteration of the second precursor was the introduction of ontology as a conceptual framework for my research into the problem of identity and privacy.

Following this false start a somewhat satirical approach was taken by converting an obselete Sensor Markup Language into an Identity Markup Language suitable for use by a Human Entity Device (see Appendix D). This was used to examine how a machine readable language might be

developed for transmission via tokens as a way of protecting human privacy, however this approach also seemed to deivate from the problem domain by appearing too specific and concrete.

To return to the lab idea of *make strange,* I reexamined a key aspect of my thesis: a public space filled with IoT devices that interact with us as we move through it. One of the latest technologies to exploit this idea is Google's Physical Web which extends the Bluetooth Low Energy (BTLE) protocol and adds a url packet that can invoke a notification on a mobile phone as soon as the user is in range. To demonstrate this a pair of shoes had a Raspberry Pi development board hidden inside to perform the tasks that a much smaller BTLE beacon would. The dev board was programmed to broadcast a bluetooth packet that advertised a url hosted on my server that included a webpage containing information about the shoe.

The response of this demonstration in the lab prompted me to ask: *how is this of concern* and *why does this matter to me*. After examining various aspects at a technical level I was left with the simple and finite requirement which was to be able to *demonstrate* how my research is of concern. This led to thinking about the artefact and how it could generate a response not just from the lay person but also from me, the builder and programmer. Critical design theory proved to be a substantial lead in the right direction and this eventuated at the theory of Critical Engineering.

This theory is now allowing me to directly examine how issues can be explored and presented and has resulted in a direct build and commencement of examining the relationship between an IoT

device and a mobile phone (see Appendix E). There are still several weeks of research into this required before any conclusions about its usefulness can be drawn and whether or not this precursor can become a prototype for an artefact that is relevant to my thesis.

Conclusion

"I am utterly amazed, utterly enchanted. I have a precursor, and what a precursor!" (Nietzsche 92). The precursor has now evolved from a snip of server-side code that posts a Facebook status update to a critically engineered piece of technology and code that seeks to provoke a response that will assist in researching the central question of my thesis: how has the idea of privacy evolved over time and what role might it play within the context of an IoT environment? It has been a long twelve week journey which has resulted in significant refinements to both the content and methodology of my research problem.

Works Cited

Kramer, Adam DI, Jamie E. Guillory, and Jeffrey T. Hancock. "Experimental Evidence of Massive-Scale Emotional Contagion through Social Networks." PNAS 111.29 (2014): 10779. Print.

Nietzsche, Friedrich Wilhelm, and Walter Arnold. Kaufmann. The Portable Nietzsche. New York: Penguin, 1976. Print.

Turing, Alan M. "Computing Machinery and Intelligence." Mind 59.236 (1950): 433–460. Print.

Appendix A

(index.php - facebook status post web server code)

```
<?php

        require 'src/config.php';

        require 'src/facebook.php';

        $facebook = new Facebook(array(

                'appId' => $config['App_ID'],

                'secret' => $config['App_Secret'],

                'cookie' => true

        ));

        $publish = $facebook->api('me/feed/', 'post',

                array('access_token' => $params['access_token'],

                'message' => 'First they came for the $identity_token and I did not

                        speak out - because I am not a $identity_token.',

                'from' => $config['App_ID']

        ));

        $message = 'Default status updated.<br>';

?>
```

Appendix B

(index.php - poetry bot web server poem creation and display code)

```
<?php

header('Content-Type: text/html; charset=utf-8');

$version = "4.2";

echo "<!DOCTYPE html PUBLIC \"-//W3C//DTD XHTML 1.0 Strict//EN\" \"http://www.w3.org/TR/xhtml1/DTD/xhtml1-
strict.dtd\">\n";

echo "<html xmlns=\"http://www.w3.org/1999/xhtml\"xml:lang-\"en\" lang-\"en\"><head>\n";

echo "<title>Poetry by me</title>\n";

echo "<link href=\"poetry.css\" rel=\"stylesheet\" type=\"text/css\" />\n";

echo "<script src=\"http://ajax.googleapis.com/ajax/libs/jquery/1.11.1/jquery.min.js\"></script>\n";

echo "</head>\n";

echo "<body>\n";
```

```
echo "<h1>Hello World!</h1>\n";

echo "<p><strong><pre>I will write some poetry and I will use stanzas of words.</strong></pre>\n";

$numberWords = mt_rand(15, 20);

echo "<strong><pre>I could use " . $numberWords . " words per stanza.</pre></strong>\n";

$numberChars = mt_rand(75, 100);

echo "<strong><pre>I could use a total of " . $numberChars ." characters per stanza.</pre></strong>\n";

$stanzaSize = 4;

echo "<strong><pre>I could use " . $stanzaSize . " lines per stanza.</pre></strong>\n";

function recallWordList($letterCount) {

  $filename = "vocab/" . $letterCount . ".txt";

  return file_get_contents($filename);

}

echo "<pre>";

for ($i = 2; $i <= 8; $i++) {

  $count = 0;

  $count = str_word_count(recallWordList($i));

  echo "I remember " . $count . " " . $i . "-letter words\n";

}

echo "</pre>";

$turingFile = "vocab/turing.txt";

$turingCount = str_word_count(file_get_contents($turingFile));

echo "<strong><pre>I will also use <a href=\"vocab/turing.txt\" target=\"_blank\">Turing's</a> " . $turingCount .
" word paper as source of words.</pre></strong>\n";

$contagionFile = "vocab/contagion.txt";

$contagionCount = str_word_count(file_get_contents($contagionFile));

echo "<strong><pre>As well as the Facebook emotional contagion experiment " . $contagionCount . " word <a
href=\"vocab/contagion.txt\" target=\"_blank\">paper</a>.</pre></strong>\n";

$wordsPerLine = $numberWords / $stanzaSize;

$charsPerLine = floor($numberChars / $stanzaSize);

function getPattern($pattern, $init, $total) {

  $remainder = 0;

  switch ($pattern) {
```

```
    case 0:
      $remainder = $total - ($init + ($init + 1) + ($init - 2));
      if ($remainder <= 0) $remainder = 2;
      $wordIndex = array($init, $init - 1, $init + 2, $remainder);
      break;
    case 1:
      $remainder = $total - ($init + ($init - 1) + ($init + 2));
      if ($remainder <= 0) $remainder = 2;
      $wordIndex = array($init, $init - 1, $init + 2, $remainder);
      break;
    case 2:
      $remainder = $total - ($init + ($init + 2) + ($init - 1));
      if ($remainder <= 0) $remainder = 2;
      $wordIndex = array($init, $init - 1, $init + 2, $remainder);
      break;
    case 3:
      $remainder = $total - ($init + ($init - 2) + ($init + 1));
      if ($remainder <= 0) $remainder = 2;
      $wordIndex = array($init, $init - 1, $init + 2, $remainder);
      break;
    default:
  }
  return $wordIndex;
}
function pickListWord($fileOfWords) {
  $count = str_word_count($fileOfWords);
  $list = preg_split('/\s+/', $fileOfWords);
  $randomWordNumber = mt_rand(0, $count - 1);
  return $list[$randomWordNumber];
}
function pickTuringWord() {
```

```
  $turingFile = "vocab/turing.txt";
  $fileOfWords = file_get_contents($turingFile);
  $count = str_word_count($fileOfWords);
  $list = preg_split('/\s+/', $fileOfWords);
  $randomWordNumber = mt_rand(0, $count - 1);
  return $list[$randomWordNumber];
}
function pickContagionWord() {
  $contagionFile = "vocab/contagion.txt";
  $fileOfWords = file_get_contents($contagionFile);
  $count = str_word_count($fileOfWords);
  $list = preg_split('/\s+/', $fileOfWords);
  $randomWordNumber = mt_rand(0, $count - 1);
  return $list[$randomWordNumber];
}
function stanzaPoem($charsPerLine, $wordsPerLine, $loopRun) {
  $poem = "";
  $stanzaSize = 4;
  $initCount = floor($charsPerLine / $wordsPerLine);
  $initCount += mt_rand(0, 2);
  for ($loop = 0; $loop < $loopRun; $loop++) {
    for ($lines = 0; $lines < $stanzaSize; $lines++) {
      $pattern = getPattern($lines, $initCount, $charsPerLine);
      for ($words = 0; $words < $wordsPerLine - 1; $words++) {
          if (($lines % 2 == 0) && ($words == 0)) {
            $poem .= ucfirst(pickListWord(recallWordList($pattern[$words])));
          }
          else if (($lines >= 3) && ($words >= 3)) {
            $poem .= pickListWord(recallWordList($pattern[$words]));
          }
          else {
```

```
            if (rand(0, 1) == 1) {
              $poem .= lcfirst(preg_replace("/[^A-Za-z0-9 ]/", '', pickTuringWord()));
            }
            else {
              $poem .= lcfirst(preg_replace("/[^A-Za-z0-9 ]/", '', pickContagionWord()));
            }
          }
        if ($words <= $wordsPerLine - 2) {
          $poem .= " ";
        }
      }
      $poem = rtrim($poem, " ");
      if ($lines % 2 == 0) {
        $poem .= ",";
      }
      else {
        $poem .= ".<br />";
      }
      $poem .= "<br />\n";
    }
  }
  //writePoem($poem);
  return $poem;
}
function writePoem($poem) {
  $filename = "buffer.txt";
  if (file_exists($filename)) {
      if (filesize($filename) >= 1000000) {
        echo "max size reached.<br />\n";
        return;
      }
```

```
  }
  $poem = filter_var($poem, FILTER_SANITIZE_STRING);
  $fileHandle = fopen($filename, "a") or die("Unable to open file!");
  fwrite($fileHandle, $poem);
  fclose($fileHandle);
}
echo "<strong><pre>My poem:</pre></strong>\n";
echo "<strong><pre><div class=\"poem\">\n";
$poem = stanzaPoem($charsPerLine, $wordsPerLine, 1);
echo $poem;
echo "</div></pre></strong>\n";
// gcm.php
echo "<form target=\"_blank\" method=\"get\" action=\"\">
    <button type=\"submit\">no</button></form><br /><br />\n";

echo "version: " . $version . "<br />\n";
echo "<em><pre>eof.</pre></em><br />\n";
echo "</p>\n";
echo "</body></html>";
?>
```

Appendix C

(synth.js - poetry bot web server audio synthesis code)

```
var body = document.querySelector('body');
var audioCtx = new (window.AudioContext || window.webkitAudioContext)();
var oscillator = audioCtx.createOscillator();
var masterVolume = audioCtx.createGain();
var oscillator = audioCtx.createOscillator();
var baseFreq = 3000;
var freqMultiplier = 100;
var curentFreq = baseFreq;
```

```
var poeticFreq = "";

var frequencyArray = new Array();

var freqArrayLength;

var freqCounter = 0;

var freqNumber = 0;

var intervalPlay;

var render = document.querySelector('.render');

var refresh = document.querySelector('.refresh');

var inputter = document.querySelector('.inputter');

meSpeak.loadConfig("mespeak_config.json");

meSpeak.loadVoice('en-us.json');

masterVolume.gain.value = 0.05;

oscillator.connect(masterVolume);

masterVolume.connect(audioCtx.destination);

oscillator.type = 'sine';

oscillator.frequency.value = baseFreq;

function getCharFreq(letterRequest) {

  switch (letterRequest) {

    case 'a':

      return 18000;

    case 'b':

      return 18075;

    case 'c':

      return 18150;

    case 'd':

      return 18225;

    case 'e':

      return 18300;

    case 'f':

      return 18375;

    case 'g':
```

```
  return 18450;
case 'h':
  return 18525;
case 'i':
  return 18600;
case 'j':
  return 18675;
case 'k':
  return 18750;
case 'l':
  return 18825;
case 'm':
  return 18900;
case 'n':
  return 18975;
case 'o':
  return 19050;
case 'p':
  return 19125;
case 'q':
  return 19200;
case 'r':
  return 19275;
case 's':
  return 19350;
case 't':
  return 19425;
case 'u':
  return 19500;
case 'v':
  return 19575;
```

```
    case 'w':
      return 19650;
    case 'x':
      return 19725;
    case 'y':
      return 19800;
    case 'z':
      return 19875;
    }
    return 200;
}
function loadCharSequence(charSequence) {
  var charFreq = "";
  var character = "";
  var frequencyString = "";
  frequencyArray = new Array();
  for (var i = 0; i < charSequence.length; i++) {
    character = charSequence.charAt(i);
    charFreq = getCharFreq(character);
    if (charFreq == 200) {
      frequencyString += "<br />\n";
    }
    else {
      frequencyArray.push(charFreq);
      frequencyString += charFreq + " ";
    }
  }
  // not this as tto long
  //document.querySelector('.freqchar').innerHTML = frequencyString + "<br />";
  intervalPlay = setInterval(playCharSequence, 100);
}
```

```
function playCharSequence() {
  if (freqCounter >= frequencyArray.length) {
    stopIntervalPlay();
    return;
  }
  else {
    freqNumber = Number(frequencyArray[freqCounter]);
    document.querySelector('.display').innerHTML = "frequency: " + freqNumber + " hz";
    oscillator.frequency.value = freqNumber;
    freqCounter += 1;
  }
}
function stopIntervalPlay() {
  clearInterval(intervalPlay);
  intervalPlay = 0;
  masterVolume.disconnect(audioCtx.destination);
  render.setAttribute('data-state', "false");
  render.innerHTML = "render";
}
function parsePoem() {
  var htmlRegex = /(<([^>]+)>)/ig;
  var poemString = document.querySelector('.poem').innerHTML;
  var stripped = poemString.replace(/^[-\w\s]+$/gi, '');
  stripped = stripped.replace(htmlRegex, "");
  stripped = stripped.toLowerCase();
  console.log(stripped);
  return stripped;
}
function textNodesUnder(el) {
  var n;
  var a = [];
```

```
  var walk = document.createTreeWalker(el, NodeFilter.SHOW_TEXT, null, false);
  while(n = walk.nextNode()) {
      a.push(n.textContent);
  }
  return a;
}
/*
render.onclick = function() {
  if (render.getAttribute('data-state') === "false") {
    render.setAttribute('data-state', "true");
    render.innerHTML = "playing";
    oscillator.start();
    renderPoem();
    renderMespeak();
    masterVolume.connect(audioCtx.destination);
  }
  else {
    masterVolume.disconnect(audioCtx.destination);
    render.setAttribute('data-state', "false");
    render.innerHTML = "render";
  }
}
*/
/*
refresh.onclick = function() {
  if (refresh.getAttribute('data-state') === "false") {
    document.location.reload(true);
  }
}
*/
/*
```

```
inputter.onclick = function() {

  window.open("./inputter.html", "_blank", "toolbar=no, scrollbar=no, resizable=yes, top=0, left=0, width=400,
height=200");



}

*/

function renderMespeak() {

  //

  console.log("render speech");

  var speech = textNodesUnder(document.documentElement);

  console.log(speech);

  meSpeak.speak(speech);

  //meSpeak.stop(id);

}

function renderPoem() {

  freqCounter = 0;

  poeticFreq = parsePoem();

  loadCharSequence(poeticFreq);

}
```

Appendix D

(human_entity_device.xml - markup language with referenced schema and stylesheet)

```
<?xml version="1.0" encoding="UTF-8"?>

<?xml-stylesheet type="text/xsl" href="iml/1.0/hed.xsl"?>

<iml:PhysicalComponent iml:id="Human_Entity_Device"

  xmlns:iml="http://www.akm.net.au/rmit/honours/sensor/iml/"

  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

  xmlns:xlink="http://www.w3.org/1999/xlink"

  xsi:schemaLocation="http://www.akm.net.au/rmit/honours/sensor/iml/1.0">

  <!-- ============================================== -->

  <!--                  System Description               -->
```

```
<!-- ================================================ -->
<iml:description>
  A first construction for a human entity device (mobile phone)
  schema that tokenises identity and provides for privacy via
  0 == NOT NULL where an acceptable value is 0.
</iml:description>
<!-- ======================================== -->
<!--               Lifespan                 -->
<!-- ======================================== -->
<iml:component>
    <iml:definition>http://www.akm.net.au/rmit/honours/iml-ont/iml-ont-1#Lifespan</iml:definition>
    <iml:classifier>
      <iml:definition>http://www.akm.net.au/rmit/honours/iml-ont/iml-ont-1#Timestamp</iml:definition>
      <iml:label>Time</iml:label>
      <iml:value>2016-04-15T16:02:10+00:00</iml:value>
    </iml:classifier>
    <iml:classifier>
      <iml:definition>http://www.akm.net.au/rmit/honours/iml-ont/iml-ont-1#Terminate</iml:definition>
      <iml:label>Terminate</iml:label>
      <iml:value>23:59</iml:value>
    </iml:classifier>
</iml:component>
<!-- ======================================== -->
<!--               Identifiers              -->
<!-- ======================================== -->
<iml:component>
    <iml:definition>http://www.akm.net.au/rmit/honours/iml-ont/iml-ont-1#Identifier</iml:definition>
    <iml:classifier>
      <iml:definition>http://www.akm.net.au/rmit/honours/iml-ont/iml-ont-1#LongName</iml:definition>
      <iml:label>Long Name</iml:label>
      <iml:value>String</iml:value>
```

```
    </iml:classifier>
    <iml:classifier>
      <iml:definition>http://www.akm.net.au/rmit/honours/iml-ont/iml-ont-1#ShortName</iml:definition>
      <iml:label>Short Name</iml:label>
      <iml:value>String</iml:value>
    </iml:classifier>
    <iml:classifier>
      <iml:definition>http://www.akm.net.au/rmit/honours/iml-ont/iml-ont-1#BirthDate</iml:definition>
      <iml:label>Birth Date</iml:label>
      <iml:value>Integer</iml:value>
    </iml:classifier>
    <iml:classifier>
      <iml:definition>http://www.akm.net.au/rmit/honours/iml-ont/iml-ont-1#BirthPlace</iml:definition>
      <iml:label>Birth Place</iml:label>
      <iml:value>String</iml:value>
    </iml:classifier>
</iml:component>
<!-- ====================================== -->
<!--               Biology                  -->
<!-- ====================================== -->
<iml:component>
    <iml:definition>http://www.akm.net.au/rmit/honours/iml-ont/iml-ont-1#Biological</iml:definition>
    <iml:classifier>
      <iml:definition>http://www.akm.net.au/rmit/honours/iml-ont/iml-ont-1#DNA</iml:definition>
      <iml:label>DNA</iml:label>
      <iml:value>Object</iml:value>
    </iml:classifier>
    <iml:classifier>
      <iml:definition>http://www.akm.net.au/rmit/honours/iml-ont/iml-ont-1#Family</iml:definition>
      <iml:label>Family</iml:label>
      <iml:value>Object</iml:value>
```

```
    </iml:classifier>
    <iml:classifier>
      <iml:definition>http://www.akm.net.au/rmit/honours/iml-ont/iml-ont-1#Physical</iml:definition>
      <iml:label>Physical</iml:label>
      <iml:value>Object</iml:value>
    </iml:classifier>
</iml:component>
<!-- ====================================== -->
<!--               Personality              -->
<!-- ====================================== -->
<iml:component>
    <iml:definition>http://www.akm.net.au/rmit/honours/iml-ont/iml-ont-1#Personality</iml:definition>
    <iml:classifier>
      <iml:definition>http://www.akm.net.au/rmit/honours/iml-ont/iml-ont-1#Culture</iml:definition>
      <iml:label>Culture</iml:label>
      <iml:value>Object</iml:value>
    </iml:classifier>
    <iml:classifier>
      <iml:definition>http://www.akm.net.au/rmit/honours/iml-ont/iml-ont-1#Politics</iml:definition>
      <iml:label>Politics</iml:label>
      <iml:value>Object</iml:value>
    </iml:classifier>
    <iml:classifier>
      <iml:definition>http://www.akm.net.au/rmit/honours/iml-ont/iml-ont-1#Religion</iml:definition>
      <iml:label>Religion</iml:label>
      <iml:value>Object</iml:value>
    </iml:classifier>
    <iml:classifier>
      <iml:definition>http://www.akm.net.au/rmit/honours/iml-ont/iml-ont-1#Sexuality"</iml:definition>
      <iml:label>Sexuality</iml:label>
      <iml:value>Object</iml:value>
```

```
    </iml:classifier>
</iml:component>
<!-- ====================================== -->
<!--               Membership               -->
<!-- ====================================== -->
<iml:component>
    <iml:definition>http://www.akm.net.au/rmit/honours/iml-ont/iml-ont-1#Membership</iml:definition>
    <iml:classifier>
      <iml:definition>http://www.akm.net.au/rmit/honours/iml-ont/iml-ont-1#Group</iml:definition>
      <iml:label>Membership</iml:label>
      <iml:value>Object</iml:value>
    </iml:classifier>
</iml:component>
<!-- ====================================== -->
<!--               Reputation               -->
<!-- ====================================== -->
<iml:component>
    <iml:definition>http://www.akm.net.au/rmit/honours/iml-ont/iml-ont-1#Reputation</iml:definition>
    <iml:classifier>
      <iml:definition>http://www.akm.net.au/rmit/honours/iml-ont/iml-ont-1#Law</iml:definition>
      <iml:label>Law</iml:label>
      <iml:value>Object</iml:value>
    </iml:classifier>
    <iml:classifier>
      <iml:definition>http://www.akm.net.au/rmit/honours/iml-ont/iml-ont-1#Business</iml:definition>
      <iml:label>Business</iml:label>
      <iml:value>Object</iml:value>
    </iml:classifier>
    <iml:classifier>
      <iml:definition>http://www.akm.net.au/rmit/honours/iml-ont/iml-ont-1#Civic</iml:definition>
      <iml:label>Civic</iml:label>
```

```
        <iml:value>Object</iml:value>
      </iml:classifier>
  </iml:component>
  <!-- ====================================== -->
  <!--               Agency                   -->
  <!-- ====================================== -->
  <iml:component>
    <iml:definition>http://www.akm.net.au/rmit/honours/iml-ont/iml-ont-1#Agency</iml:definition>
      <iml:classifier>
        <iml:definition>http://www.akm.net.au/rmit/honours/iml-ont/iml-ont-1#Record</iml:definition>
        <iml:label>Record</iml:label>
        <iml:value>Object</iml:value>
      </iml:classifier>
      <iml:classifier>
        <iml:definition>http://www.akm.net.au/rmit/honours/iml-ont/iml-ont-1#Accessor</iml:definition>
        <iml:label>Accessor</iml:label>
        <iml:value>Object</iml:value>
      </iml:classifier>
  </iml:component>
</iml:PhysicalComponent>
```

Appendix E

(precursor2_final-lamp.jpg – image of precursor 2 lamp and mobile phone as a critical object)