Foundations of Software Architecture

Assignment Guide

Spring 2005

Authors
John Reekie, University of Technology, Sydney

Contributors and reviewers
Lian Loke, University of Technology, Sydney
Rohan McAdam, Honeywell Inc.

Terms of Use

Contents

This document contains all information needed to perform the assignments in this subject. The assignments are worth a total of **60** marks, so make sure you read this part carefully!

Note that assignments are to be implemented using only approved technologies. By default, you are expected to use Java, to use the Eclipse development platform and the provided source-code repository, and to use the third-party products used in the Labs. One of the key issues in marking your programming deliverable is that it is easy to build and run your system, so we have to limit the number of different packages that are being used by the class.

1 Conduct and deliverables

There are two assignments, which are undertaken in parallel:

- The written assignment produces the Software Architecture Document.
- The lab assignment produces the Executable Architectural Prototype.

These two artifacts are defines by the Unified Process, which we are (roughly) following in this subject. Each is delivered in three iterations, each terminated by a milestone with a specified set of deliverables. Marks are awarded for meeting each milestone; in addition, the final version of each assignment is marked.

1.1 Milestones

Three milestones are set throughout the semester. The milestones for each assignment are explained carefully in Sections 3, 4, and 5—make sure that you read the requirements for each milestone carefully! Note also that:

- Milestones are due at the specified dates and times given in the Subject Guide. There are **no extensions** for delivery of a milestone. If you do not deliver on time, you will not receive the mark for that milestone.
- The milestones for both assignments occur *at the same time*. Do not expect to be able to deliver one on time, and the other late. If you do, you will only receive marks for the one that you delivered on time.
- At some of these milestones, you will be performing a peer-review inspection with another team. You will then have a week to address and correct the issues noted by the inspection team.

1.2 Final deliverables

The final deliverables are due at the dates and times specified in the Subject Guide. There will be no extensions given for late delivery.

The final deliverables are assessed on work performed *over and above* the minimum criteria specified for each milestone.

1.3 Teams

All assignment work is performed in teams of **three**. You must form yourself into teams by the end of the **second week** of semester. If you are not in a team by then, you will be assigned to one.

Each milestone includes a personal reflection by each team member on the work conducted for that milestone. If the instructor feels that your reflection indicates lack of participation, you will be required to re-do and re-submit that milestone and all subsequent milestones individually.

2 Final deliverables

2.1 Software Architecture Document

The Software Architecture Document describes the design of the architecture, and provides the basis on which further design and implementation work takes place. It should:

- Summarize the system's purpose
- Analyze the system context
- Describe elaborated customer needs as a set of usage and quality narratives.
- Provide a set of views that describe the architecture, including its structure, behavior, implementation, constraints, and so on.
- Justify all architectural decisions. Refer to the system context, stakeholder input, results of prototypes, and so on.

The Software Architecture Document's primary purpose is to enable further development in the Construction phase. The document will therefore be assessed on how well it fulfills that purpose. To that end:

- Make sure that the document is no longer or more verbose than needed. Remember that the document will be read (amongst others) by impatient software developers. You will be marked down severely for space-fillers or excess verbosity.
- Make sure that the document and your design are realistic. Don't design some fancy pie-in-the-sky architecture that the software development teams can't or won't build. Everything in the architecture must be feasible, and that feasibility must be supported by reasoning or prototypes.

2.2 Executable Prototype

The executable prototype is the core of the system being built. It provides the platform on which subsequent development takes place (in the Construction phase). It should therefore:

- Contain all key pieces of the architectural infrastructure.
- Implement key pieces of functionality that prove the architecture's feasibility and act as a model for further development.
- Have resolved all configuration management issues.

The executable prototype must represent the architecture expressed in the Software Architecture Document (and vice versa). Any discrepancies between the two will reduce your marks severely. The quality of your configuration management and the readability of your source code also play a big part in your final mark.

2.3 Assessment

The Software Architecture Document and the Executable Prototype will be assessed on their suitability as the basis of the Construction phase of the development lifecycle. The document and the source code will be read from the viewpoint of a Software Development team leader, who has to ensure that the Construction and Transition phases result in a successful working system.

As such, the most important thing to remember is that you are not being marked on the *quantity* or the *complexity* of your work. On the contrary, you will be marked higher for *brevity*, *clarity*, and *simplicity*. This applies to the written documents as well as the source code.

Exceptional merit

In exceptional circumstances, the instructors may award an additional five bonus marks to an individual or team for outstanding quality of the final deliverables.

3 Milestone 1

For this milestone, you are expected to produce a lightweight set of Lifecycle Objectives, including the proposed conceptual architecture of the system, and to make sure that you are capable of performing the necessary software development tasks for this project.

3.1 Conceptual Architecture

This is the written assignment for this milestone.

Minimum requirements

You must deliver the following to meet this milestone:

1. An initial system context description

   Identify four CERTs (factors or forces) relevant to this system, and write a short paragraph on each. One must be from each of the four categories (market, technological, policy, and organizational).

2. An initial usage model

   Write four short usage narratives for this system.

3. An initial conceptual architecture

   Print out your system description, and with a pencil or highlighter, identify key concepts. Use those concepts to create an initial set of conceptual components, and then an initial conceptual architecture.

4. An elaborated conceptual architecture

   Refine the conceptual architecture. Mark each component with a stereotype to indicate its nature.

5. An initial behavioral analysis

   Identify four events from your usage narratives, and draw a use-case map for each on the conceptual architecture. Identify responsibilities at each crossing.

Aiming higher

If you are planning to obtain higher than a pass mark in this subject, you should also deliver some or all of:

- A more thorough system context
- A greater number of more refined usage narratives
- Analysis and refactoring of the conceptual architecture to account for anticipated quality attribute needs
- Analysis and modification of the conceptual architecture to improve usability

3.2 Initial Development Capability

This is the programming assignment for this milestone.

Minimum requirements

You must deliver the following to meet this milestone:

1. Create the directory structure that you will need for this project, which includes the code from Lab 1 as a demonstration prototype.
2. Demonstrate to the instructor that you are able to use the Eclipse platform to:
   (a) Check out your development tree.
   (b) Modify the ThreadedServer program, compile, and run it.
   (c) Check your changes back into the development tree.

Aiming higher

If you are planning to obtain higher than a pass mark in this subject, you should also deliver some or all of:

- A user interface technical prototype

  Create a simple Java GUI that communicates with a program based on the ThreadedServer program from Lab 1. The GUI should contain some buttons or other simple user interface widgets to exercise the events identified in your conceptual architecture, and to display responses from your modified ThreadedServer program.

3.3 Individual reflection

Deliver a personal reflection on this milestone. This must be done individually. You should describe what you learnt, and how, during the conduct of this milestone. The reflection should be half to one page (no more) in length.

4 Milestone 2

The purpose of this milestone is to design, explore, and validate the execution architecture.

4.1 Execution Architecture Design

This is the written assignment for this milestone.

Minimum requirements

You must deliver the following to meet this milestone:

1. A concurrent subsystems view

   Create an initial concurrent subsystems view of the system. For each component, give the mapping from the conceptual architecture. Identify stereotypes and connector styles. Provide an explanation of the architecture, including a short description of each concurrent component, explaining why it is a separate component.

2. An initial implementation architecture

   Create an initial implementation diagram of one of two key sections of the system.

3. Quality attribute analysis

   Identify four quality-attribute-related needs and the corresponding architectural responses. Write a paragraph on each, explaining why they are significant to *this* system.

4. Behavioral analysis

   Create use-case maps on the execution and implementation architectures.

Aiming higher

If you are planning to obtain higher than a pass mark in this subject, you should also deliver some or all of:

- A deployment view
- A more detailed concurrency view
- Consideration and analysis of off-the-shelf infrastructure.
- Your own research into the way systems like this are used.

4.2 Execution Architecture Validation

This is the programming assignment for this milestone.

Minimum requirements

You must deliver the following to meet this milestone:

1. A running system that contains the key elements of your execution architecture. Usually, the most straight-forward way to achieve this is to start a separate process for each concurrent subsystem.

   Minimum requirements to pass this milestone:

   - The system must be able to be started with a single command. Although real systems are often not like this, for the purposes of ease of verification and marking, it is a requirement for your prototype.
   - It must be possible to start a telnet session to connect to the system, and enter the name of four different events identified in your architectural document. When entering these commands, each concurrent subsystem must print out the key areas of responsibility that is being exercised. (If you wrote a GUI test harness, you can skip this part).
   - The code *must* match the execution architecture design. If your design has 18 concurrent subsystems, then so must your program. (This is a powerful incentive to make your design no more complex than necessary.)

Aiming higher

If you are planning to obtain higher than a pass mark in this subject, you should also deliver some or all of:

- Use a GUI front-end to exercise the system.
- Use JMeter or some other profiling tool to characterize some aspect of the system's performance.

4.3 Individual reflection

Deliver a personal reflection on this milestone. This must be done individually. You should describe what you learnt, and how, during the conduct of this milestone. The reflection should be half to one page (no more) in length.

5 Milestone 3

This milestone refines and completes the design started in the previous iteration, and completes the executable architectural prototype.

5.1 Architectural Elaboration

This is the written assignment for this milestone.

Minimum requirements

You must deliver the following to meet this milestone:

1. A refined execution architecture

   Elaborate and expand the execution architecture to produce two views: a concurrent subsystems view, and a more detailed concurrency view.

2. A refined implementation architecture

   Identify application and infrastructure components, and identify the nature of each infrastructure component (product-specific, product-line, or off-the-shelf). Briefly describe the interfaces to all components.

3. A refined conceptual architecture

   Modify the conceptual architecture to account for changes and new understanding created by the development of the execution and implementation architecture.

4. Behavior analysis

   Four use-case maps and four impact maps on the relevant views of the architecture.

Aiming higher

If you are planning to obtain higher than a pass mark in this subject, you should also deliver some or all of:

- A complete quality-response analysis.
- More comprehensive behavioral analysis.
- Decisions about off-the-shelf infrastructure.

5.2 Executable Prototype

This is the programming assignment for this milestone.

Minimum requirements

You must deliver the following to meet this milestone:

1. A completed architectural prototype. This prototype forms the basis of the Construction phase (of the Unified Process), and at a minimum must:
    - Compile and build with a single *ant* command.
    - Launch with a single command (usually, some form of startup script).
    - Exhibit skeleton functionality for two complete usage narratives.

Aiming higher

If you are planning to obtain higher than a pass mark in this subject, you should also deliver some or all of:

- One or more technical prototypes.
- Skeleton functionality for additional usage narratives. The narratives should cover the range of system functions, to establish a high degree of confidence that the architecture is capable of meeting its functional needs.
- Include additional off-the-shelf infrastructure. (Your written report must justify the need for this infrastructure with a CERT analysis.)
- Testing and monitoring capabilities.

5.3 Individual reflection

Deliver a personal reflection on this milestone. This must be done individually. You should describe what you learnt, and how, during the conduct of this milestone. The reflection should be half to one page (no more) in length.

A Software systems narratives

All assignments are based around one of a small set of software systems provided by the instructors.

A small number descriptions of software systems will be provided early in the semester. Each is a "vision" level description of the system, from which you will need to extract requirements as needed. Don't get hung up about not having detailed requirements specifications, just approach the problem with flexibility and an open mind and you'll be fine.

Bear in mind also that the purpose of these system descriptions is simply to provide you with a starting point and focus for exploring the field of software architecture. You won't be penalized for embellishing or altering your system, provided that you're reasonable and consistent about it.

A.1 Custom Shooz

(This is a short system narrative that can be used in the tutorial.)

Custom Shooz Co Ltd, of Seattle WA, has built up a solid business over several years of careful growth and customer service. Their business is making custom shoes, and they have a range of cool designs that they customize to customers' wants and (of course) feet. Having established skills and a solid client base locally, they are now interested in a significant expansion of the business, and want to do it with the aid of a web site that enables them to take care of their customers remotely.

The President of Custom Shooz, Funk O. Sole, explains:

"Well, we've always had requests from out-of-towners for custom work. If they're an existing customer, it's fine, because we keep the template of the foot and other measurements here. Otherwise, it's kinda difficult, because one of the things we pride ourselves on is giving people a better fit than they can ever get off the rack. And with a much cooler, better-looking shoe as well!

"So, what we did was develop a little measurement kit that we send out to folks and they have to send it back. We've improved it over the last couple of years so that it's almost foolproof. Once we have the customer's measurement kit in, we can produce almost any shoe from our range—all the custom stuff, like stitched-on patterns, dye colors and finishes, laces and buckles, can be done without them ever being within a thousand miles of our store!"

Custom Shooz plan to advertise using conventional means, but want the website to be a location where customers can find out about their custom range, get the measurement kit, and customize and order shoes. They also want the site to interface to their accounting system.

A.2 Whale Early Warning System

Howard Hugs, a rich Australian entrepreneur-turned-conservationist, is outraged by the Japanese government's plan to kill humpback whales in Australian waters and in the Southern Ocean whale sanctuary. Rather than pursue diplomatic means of resolving the problem, he has decided to build and deploy a warning system that governments and other concerned organizations can use to try and prevent the slaughter of humpbacks. This system, the Whale Early Warning System (WEWS), will track whales and whaling vessels, and try and predict encounters between them. Howard knows that the technology to do this is only just "there," but he is determined to try anyway. He is even prepared to pay for it by selling his garage full of Rolls-Royces and driving a Volkswagen instead.

Howard has already had some initial work done on the architecture of the system. You ask him about it.

"Well, we've got two sources of data for tracking the whales and the ships. We call the ships the 'enemy.' [laughs] First off, some of the whale pods have individuals tagged with radio transmitters. CSIRO has agreed to let us have a real-time feed from their tracking system, as long as we don't let anyone else have it... for obvious reasons...

"The second source of data is satellite imaging. I've negotiated a deal where we can get access to some limited real-time bandwidth—but we do pay a lot for it. So we have to maximize use of that bandwidth in two ways. First, by being careful about how we direct the cameras for those times when we have use of them. And second, we can upload program modules onto the satellite itself. Those of course have to be checked very carefully, both for security and real-time performance, but this means we can do some recognition and compression before trying to send the data.

"So, it's not perfect, but it's something. Then we have to analyze it and figure out what's a whale, what's a boat, and so on. That's pretty heavy-duty stuff, a lot of those servers will be doing nothing but crunching on that recognition stuff. Then we run the predictive algorithms and... well, we will have to see how well it goes and then spend a lot of time fine-tuning it, I guess. In-house, we'll have scientists here "helping" the automatic algorithms along and training them. Anyway, we will provide access to the data to any organization that we approve to have it... And some of it will be made public on a website as well, we want the public to know what's really going on.

"So, we're nearly up and running on the hardware side... we've got servers into the datacenter already, but we need to get the software side of things going pretty fast. That's where you come in. Interested?"

(In subsequent conversation, Howard Hugs reveals that he spent so much on WEWS that he can't afford to pay you yet, but you can be chauffeured around in one of the Rolls Royces until the eBay auction for it closes.)

A.3 Radio and Audio Technologies Ltd

Radio and Audio Technologies Ltd (RAT) are an established provider of digital audio technology to the recording and broadcasting industries. Up until now, their equipment has been integrated with many other brands of equipment in any given installation, but RAT is now seeking to produce a breakthrough product for smaller radio stations, a complete "off the shelf" digital broadcasting studio.

Based on discussions with potential customers, RAT have decided that the system consists of two parts. The first is a digital audio processing system, configurable with from 8 to 24 audio channels. Each channel selects from a range of sources, including CD players, microphones and voice (telephone) circuits, and digital storage media, and is able to perform a standard set of audio processing functions, such as parametric equalization, gain, and panning between two output channels. Additional processing, such as compression and noise reduction, needs to be able to be "patched" into specific input or output channels, with the number of such processing modules determined by the amount of processing power the customer buys.

The second part is the command and control system, which the operator uses to configure most of the operation of the studio. This comes in different flavors, with the lowest-cost option being fully virtual, with all controls implemented on standard PC hardware. Although many recording studios won't buy digital processing unless it has knobs on it, RAT figure that for this product, cash-pressed radio stations will accept touch-screens as a compromise.

However, RAT's standard user interface hardware will also be available for customers that can afford it. They expect to produce a mixture of configurations with varying degrees of physical hardware and screens for display and visualization.

As always, RAT want to make sure that this product lives up to the reputation they have established for themselves, that of good-sounding digital equipment that does the job and doesn't break. They need the product to be competitive, but any corners that are cut that cause the product to fail while "live" will severely damage their other markets as well. They are also acutely aware that customer expectations include 24-hour remote access to the system, both through a web interface and a more sophisticated interface that allows a subset of the full RAT control, editing, and signal processing functionality. They have hired you to lead the architectural team for the rest of the Inception phase and the Elaboration phase of this project, with a commitment to keeping you on as project leader(s) if Construction goes ahead.

As a last-minute wrinkle, RAT inform you the day before delivering the contract that they also want to re-use significant elements of this new architecture across a whole new product line, which includes digital audio storage and processing for the film industry, and storage and analysis of security (audio) tapes in large complex installations such as banks and airports.

A.4 CyberTone Systems

CyberTone Systems is poised to make a splash into the "ubiquitous targeted advertising" market. They are vying with a number of other startups and established companies trying to gain a prominent position in this market. CyberTone believe that they have the edge, but they need architectural expertise to make their product vision a reality. Wys I. Wyg, CEO, explains:

"This is an extremely challenging project. No two ways about it, it's a tough nut. Never mind the business and legal stuff, that's been hard but I think we've got that on track now, but technically it seems to be getting more complex every day. We need to put a lid on it, scope it into something feasible that we can deliver in a short time frame and wow the public and the media... oh, and the paying customers as well. Yes but of course, we want it to look good but it has to work properly and the basic architecture has to take us through several versions. We can't afford to re-architect this thing every time, we need [thumps table] the infrastructure in place to keep one jump ahead of those guys [indicates the window and therefore, presumably, their competitors].

"You know the scenario, I expect. Seen Minority Report? Tom Cruise. Great actor. One of the best. Anyway, there's a scene where he walks along the shopping mall and the walls are talking to him. They know its him and they know what he buys, and they try to get him to buy more. Well OK, we don't say that in our press releases but hey, ... anyway, look, don't repeat that and you can do it, right? Good job, we're depending on you."

A little technical perspective is provided by Mr Con Currency, CyberTone's Chief Technical Officer: "The reason we are going to win this is because we have the pedigree. We've got the research behind us now—ground-breaking research, we learned a lot and now it's time to make the early investment a reality. We're calling this the Universal Targeted eXperience, you know technically speaking we can use this for a lot of things other than advertising...

"What our strength is in the algorithms for adapting image display and audio in response to physical movement information. Once someone is within our sensor field, we know how to figure out where the person is, and what they doing. Where they're looking, what gestures they are making. All of this information feeds into algorithms that can be configured according to how you want the screens and audio to respond to that person. We can localize audio in 3D space even.

"So it's not much of a stretch to add the information about *who* the person is. Sure, there are privacy issues but I think eventually most people will opt in. Retinal scanning is a bit of a fantasy but a simple wireless device can be worn by anyone who wants our "experience." You carry a credit card, right? Of course... well that's all the space it takes. We can pick up the code and connect to a centralized identity bank. This is what makes it targeted, of course, because now we can have the algorithms tailor its output based on information like your consumer profile, demographic, and so on. Assuming it's in a database we can access, that is."

Just before you sign the contract with Bystander Technologies, they tell you that they've decided to use the architecture that you will be designing for them for other products as well, not just advertising. "You could tell someone walking into the news-agent at the airport, that they are going to be late for their flight if they don't hurry up. Now *that* would be useful!"

A.5 Automated Warehouse Control

Warehouses Automation Inc. specialize in automated warehouse control systems. Up until now, each installation has required a good deal of custom development, increasing delivery time and increasing cost dramatically (and sometimes, unpredictably).

The warehouse control market is getting hotter, though, and upstart competitors like Warehouses Galore are on the verge of coming through with scalable, turn-key integrated systems that connect right through from the customer's web shopping cart, through stock management and ordering, through to the packaging robot onto the right mail carrier's conveyer belt. Warehouse Automation Inc have realized that they need to adapt to this new market. They have hired you as a senior software engineer and you are convinced that software architecture has a large part to play; if only you can convince management to take an architecture-based approach.

But first, you have to learn more about the systems they build. Pack Rat, an old hand in the warehouse management game, elaborates:

"Basically, we realized that most of the systems that we build fall into a pattern. Our part has traditionally been the warehouse but we're more and more hooking into the customer's other systems, and as often as not providing them as well. It's like this: stuff come in; stuff gets combined; stuff goes out. Sometimes, the same stuff goes out that comes in—that's called retail or distribution. Not quite that simple though, as you still have to combine things into one order, and they also need multi-warehouse synchronization. One order might be filled from warehouses in Nevada and Hong Kong, for instance.

"Other times, the stuff that comes in gets assembled into different stuff; that's called manufacturing. We don't deal with the actual manufacturing part of the system, but we do manage everything up to and right after it. Our systems make sure that the parts needed for assembly are there when they are needed, and that assembled units get shipped to the right place. That might be a customer, or another factory where it gets built into a more complex unit.

"Anyway, we automate most of the materials handling as well. Inbound goods are delivered on pallets, scanned, and given to the first in a set of specialized handling robots. Usually, items are barcoded for identification, but manufacturers also have dedicated handling bots that we can interface to where our systems hand over to theirs. Outbound goods—same story. The picking robots will get the stuff from where it's racked, and the packing robots will... um, pack them. Any size up to a full pallet. Then it goes manual again to load onto trucks. Sometimes the whole thing is automated if everything ships by a carrier with their own barcode system.

"Don't forget that the operators and managers need to know what's going on all the time as well. Where everything is, what's coming in and going out, and when. Plant utilization, backlogs, cost analysis, job scheduling, maintenance records and reporting, historical plant reporting... you name it! Critical paths needs to be alarmed so we don't hold things up in case something fails to show up on time. And these days they [rolls eyes] need a web interface to it all as well, because they're '24/7 even on the road' or so they claim.

"Complicated enough for you? Somehow, we are supposed to create a half-dozen different products that cover the complete span and are fully scalable as well, but have all of them sharing the same codebase. Hah!"

A.6 The National Unified Tracking Service

The National Parks and Wildlife Service (NPWS) are commissioning a new computer-based tracking system for native fauna. The essential aim of the new tracking system is to improve the accuracy and amount of information available to them, for the purposes of monitoring species decline and recovery rates and enabling earlier identification of endangered species. To improve data collection, the NPWS plan to deploy remote monitoring units to gather real-time in-the-field data about the presence and activity of species of interest. The new system, dubbed the National Unified Tracking Service (NUTS), also has the goal of unifying existing species tracking data, and forming the basis of an improved public information and education system about endangered species.

NPWS realizes this is an extremely ambitious project and are concerned about development feasibility and cost. They have asked you to join the architecture team to help ensure that the architecture of the system supports its diverse goals while remaining cost-effective to implement and maintain.

Adam Eve, NPWS scientist, explains the motivation for the real-time acquisition part of the system:

"Well, we already have a number of tracking options, like tags and so on, which we will continue to use. But there are a couple of problems, like you have to catch the animal, and they don't work on reptiles and arachnids for instance. But mainly, we don't have a good way of really localizing the data—that is, getting lots of information about which critters are where exactly and what they're doing there. So, these gizmos here (Adam taps a steel box on his desk) are going to solve that problem for us. Check it out: camera, infrared, audio, a chunky little processor and battery, all in here. Oh right, and the wireless link, we're still figuring out what the best option is for that... Anyway what we do is deploy a cluster of these "bricks" in a small area, and then download software and parameters onto it that tells it what to be looking for. There's probably one or two dozen bricks in a cluster, and we figure we'll have thirty or forty clusters in the field once we get into full swing.

"So once we have a cluster configured, we set them running. The processor ticks away until the sensors show something that might be interesting, at which point it fires into full gear and runs recognition algorithms on the audio or video. If it still looks good, it starts transmitting live audio and video—compressed of course—to the main servers. There, we do some heavy-duty recognition on the real-time data coming in to get rid of data that isn't going to be of any interest. The result gets saved and is then processed later with human help in order to figure out which data to keep, and also to cross-correlate the data streams from the different bricks in the cluster. Finally, all of this gets stored in the main species tracking database.

"Bit of a beast, eh?! Anyway the end goal is to get much better localized data on species of interest, together with all the associated date/time information and some good audio and video attached to it"

A.7 Audio Magic Ltd

Audio Magic have established themselves as a reliable provider of digital audio technology to the recording and broadcasting industries. Up until now, their equipment has been components that have integrated into more complex installations, such as stand-alone disk recorders and audio processing units. They are now investigating the market for turnkey audio systems for film post-production.

Audio Magic perceive that a niche product can be produced that addresses production of the effects and foley tracks. The foley track is used to create the sounds made by the actors in the film. Traditionally, the foley sounds are created specifically for a particular film by using various props to create sounds like shoes walking, doors slamming, and so on. This work is done by a team of foley artists and the foley editor, who puts the sounds created by the artists together into the foley track. The foley section of the system therefore has to have user interface consoles and various audio recording devices located in foley studio itself.

The sounds effects crew works a little differently to the foley crew. They are less concerned with intimate sounds made by an actor (such as footsteps and chairs creaking), and more with "big" sounds like explosions and machine noises. They make use of an extensive library of pre-recorded sounds, including some purchased from other companies. Some sound effects will also be recorded or electronically synthesized specifically for a particular film.

In either case, the system has to support construction of a complex set of digitally-recorded sounds into a complete sound-track. Although there are existing systems for this, Audio Magic perceive a significant competitive advantage to an "all in one" sound management system. As one of the features they plan to include to "leapfrog" the competition, Marketing have been busy promoting a "24x7" interface to the system—basically, this means that any of the sounds effects folks can use a web browser to upload new sound effects that they have recorded from where-ever they are by using a web browser. They will also be able to manipulate the effects and manage the library remotely.

See http://www.marblehead.net/foley/ for more explanation of the process of making the sound track of a film.

A.8 Gurgle Planet

In the year 2045, the earth's resources have been drained to the point where the ecology is widely considered to be at breaking point. What used to be the "third world" has imitated the consumption patterns of what used to be the "first world," and many scientists believe that, despite significant advances in the management of natural resources in the last 50 years, the earth is on the verge of a complete ecological collapse.

An urgent meeting has been convened to find a solution. For the first time ever, the leading world powers have agreed to radically curb their own consumption in the interests of a global solution. Your company is one of several that have been asked to proposed solutions to one part of this problem: a global resource monitoring system, that can be used by the policy makers to find out where resources are being created and consumed. You have been asked to assemble a crack team of technologists and architects to investigate and propose a solution based on your company's knowhow and existing products.

In your initial investigations, you have discovered that the problem is more complicated than you thought... natural resources are so decimated that there is perceived to be a need to manage and detect not only classes of ecosystem such as forests and agricultural zones, but also the presence of every species of plant and animal, regardless of whether it is directly used in food or other production. Water and soil quality data must be collected and analyzed. The last remaining oil reserves must be detected and mapped out, as well as all other minerals.

Consumption patterns also need to be analyzed. Since not all companies or governments are forthcoming with useful data, much of is must be obtained surreptitiously, by the used of fly-over aircraft, "smart dust" wireless sensors, and satellite imagery in the visible, infrared, and radio spectra.

Finally, waste management data needs to be obtained and tracked, in order to maximize re-use of mineral resources and minimize exposure of the remaining ecosystems to harmful toxins.

You have also realized that effective use of this information demands an effective human interface that conveys the right information at the right level of detail. As a model, you have chosen a system initially developed 40 years earlier as your vision of an effective interface that combines context, variable information display, and spatialization capabilities: Google Earth. One of the cynics on your team has dubbed your developing project "Gurgle Planet," as a pun on the phrase "going down the gurgler," and referring to the earth's ecosystem.