Everything I Ever Needed to Know About

ITS Architectures

I Learned from my Father-In-Law

Introduction

The following document seeks to explain the key components of an *ITS Architecture* using familiar language and an example that everyone (except engineers) can understand. In each of the following sections, the main components of an *ITS Architecture* will be described by way of analogy, followed by an explanation of that analogy, as it applies to *ITS Architectures*.

My Father-In-Law's Cottage

Somewhere, in the wilds of the Gatineau Hills, next to a lake, sits a rather luxurious year-round cottage. This cottage and all of its surrounding buildings were built by a rather round Italian man with an enthusiasm for do-it-yourself projects, and discount tools.

Construction began years ago, when the children were in their twenties. Enthusiasm, participation, and cooperation were high, as different projects were undertaken to cultivate the unsettled bush. First came the essentials (the outhouse), followed by a screened-in gazebo with rudimentary cooking facilities. Next came a small bunky and a dock.

Soon these facilities were subject to upgrades as people devised clever ideas to satisfy further needs. The gazebo was upgraded with a shower stall, a swing set, and second-floor "tree-house". The bunky was extended to offer more sleeping space.

Each of these projects provided comfort while the main cottage was built.

The cottage was designed to comfortably house my in-laws, and their three children.

In the years that the cottage was being built, recreational gear found storage space outdoors under any structure that would shelter it from the elements; water-skis, windsurfers, fishing rods, basketball hoops, and the like.

In more recent years a rope swing was added, a volleyball court, and a rock-climbing wall.

It has now been many years since construction began, and what has been achieved?

Fifteen years later the children are in their forties, and the grandchildren are in their teens. Having busy lives, they rarely visit, which is fortunate being that the cottage was not designed to accommodate spouses and seven teenage grandchildren.

Maintenance and repairs now consume as much time as completing the construction.

Since the cottage has been built, the gazebo and bunky have become obsolete. Furthermore, the numerous ad-hoc "upgrades" to the gazebo have compromised the durability of its structure. Never having been fully completed, these buildings have decayed, and their safety has been brought into question.

All of the recreational gear that was neatly stored under the various buildings has now grown to the point that it overflows from each nook-n-cranny, greatly resembling trash. Most of it has been badly wasted, being that it was not properly stored.

The functionality provided by the gazebo and bunky was overlooked in the cottage plans, being that these needs seemed to be provided for. Now that these buildings are unusable, this functionality is being forced on the cottage. The results: in the absence of a more appropriate place, wet snowshoes, boots and jackets are left hanging from the pool table, and guests share the living room couch with the dog.

In the absence of a garage, building materials are now scattered across the lot. The gazebo is filled with assorted wood, with other materials being stashed under tarps in different piles around the lot. Over time the tarps have become ragged and torn, wasps and other animals have found shelter amongst the materials, and much of the material has been damaged by the elements.

Consideration was given to building a new garage to address some of these issues. No sooner had the planning started, it was realized that all of the available space on the lot had been used. Tearing down an older structure was considered, but that drew criticism from family members who have grown accustomed to them, and who argued that they could be rejuvenated.

My father-in-law is now faced with the reality that most of his time and money are consumed each year trying to preserve yesterday's makeshift solutions, rather than working to realizing tomorrow's dream.

A Clean Slate

As it happens, my wife and I have purchased the lot next door. It is untouched and pristine. In the years to come, we will develop the lot. It is my determined objective to avoid the pitfalls that have consumed my father-in-law, by:

- Assessing what the cottage lot could be used for;
- Exploring the activities needed to support those uses;

- Determining what facilities are needed to accommodate those activities; and finally
- Deciding what uses we should build and plan for.

In short, we will develop a *Cottage Architecture*. The elements, and purpose of a *Cottage Architecture*, as described above, are analogous to an *ITS Architecture*.

Through the following sections, an *ITS Architecture* will be explored, using a *Cottage Architecture* as an example.

An *ITS Architecture* has four components:
- *User Services*;
- A *Logical Architecture*;
- A *Physical Architecture*; and
- *Market Packages*.

User Services

A cottage means different things to difference people. Some people use a cottage for summer barbeques, swimming, and watersports. Others use it as a base for golfing, or horseback riding. If the cottage is winterized, it can be used for skiing, snowshoeing, ski-dooing, or ice fishing. More fundamentally, there are basic living needs, such as preparing food, bathing, and such.

Roughly speaking, these are *User Services*. While not all of them will be relevant to everybody, the more that are identified, the more comprehensive the *Cottage Architecture* will be and the greater value it will provide. After all, you never know if one day the little tin-boat will be replaced by a 200hp inboard/outboard, or if your daughter is going to marry a kayaker who will want to store a 23' sea kayak in your shed. If the architecture recognizes these *User Services*, then there is a greater likelihood that they may be seamlessly accommodated, though they were unexpected.

These *User Services* are analogous to the *User Services* defined under an *ITS Architecture*.

The concept of *User Services* allows system or project definition to begin by establishing the high level services that will be provided to address identified problems and needs. The following are examples of *User Services* identified as part of Canada's *ITS Architecture*:
- Traveller Information;
- Route Guidance;
- Traffic Control; and
- Incident Management.

The more comprehensive the list of *User Services*, the more guidance the architecture can provide in terms of defining standards, and integrating systems. However, technology tends to plot its own course, and quickly, so it is impossible for the list of *User Services* to be comprehensive.

Additionally, these *User Services* have *Sub-Services*. To continue the cottage analogy, swimming might mean different things to different people. Some people might view a diving board as an essential part of swimming, others might see a snorkel and mask as critical, whereas others might consider an inflatable mattress and a drink cooler as paramount. These could each be considered as the *Sub-Services* of the *Swimming User Service*.

Canada's *ITS Architecture* identifies a *Traveller Information User Service*. This *User Service* has a number of *Sub-Services*, for example:

- Broadcast Traveller Information
- Interactive Traveller Information
- Real-time Ridesharing Information.

Logical Architecture

Consider the *Boating User Service* from the *Cottage Architecture*. Anyone who has a tin-boat can attest what a pain in the rump they are. Each spring the boat has to be put in the water. The outboard motor then has to be taken out of storage and tuned-up. It then has to be installed on the boat. Next comes the safety equipment. The gas tank needs to be filled with gas, and the appropriate amount of oil mixed in. The boat is then tied at the dock. When it rains, the boat has to be covered so it doesn't sink. Then, in the fall, the whole process is done in reverse before the lake freezes.

This is in essence a *Logical Architecture*. A *Logical Architecture* defines the processes that must be performed in support of the *User Services*. The example above is a summary of the processes required to support the *Boating User Service*. In an actual architecture, the process map would be complete, and would include all *User Services*.

An *ITS Architecture* goes further to define information or data flows that are shared between processes. This facilitates the definition of standards for communication, the information gathered, and the formats.

Physical Architecture

Spring at my father-in-law's cottage is exciting. Last year, the event of putting the boat in the water was fantastic, with emotional firework displays, and flying pots.

The boat is stored on the dock, which is convenient. However, not having a garage, boathouse or workshop, my father-in-law tuned-up the motor in the living room (on the rug). A rather grumpy raccoon was engaged to liberate the oars from under the gazebo, and the gas tank was under the porch, where it had filled with water. Consequently, the

oil and gas were mixed on the porch, where a spill left a rather nasty oil slick. The flying pot thing happened when my mother-in-law got home to find the oil slick, and the rug.

All this to say, that all of the processes that are defined under the *Logical Architecture* have to take place somewhere. The *Physical Architecture* describes where those processes take place.

A *Cottage Architecture* might provide that boating equipment is to be stored in a garage or boathouse, conveniently located next to the workshop, where the tune-ups take place. Mixing the oil with the gas is to happen on the gravel driveway, where spills are inconsequential. The boat, motor, and equipment are to be carried down the path to the dock. The actual act of launching the boat is to take place at the dock.

In an *ITS Architecture*, the *Physical Architecture* serves the same purpose in that it describes where the various processes take place. Its main components are *Subsystems* and *Terminators*.

Subsystems are physical locations and are the principle structural elements of the *Physical Architecture*, for example: traffic operations centers, roadside signal controllers, and even automobiles. There are 23 different *Subsystems* in the Canadian *ITS Architecture* and they are grouped into four classes: Centres, Wayside, Vehicles, and Travellers.

Terminators define the boundaries of an *ITS Architecture*. *Terminators* represent the people, systems, and general environment that interface with the ITS systems included in the architecture, but being that they represent the boundaries of the architecture, they are not allocated functional requirements.

To continue the cottage analogy, the road to the cottage might be best represented as a *Terminator*. The *Cottage Architecture* might include a driveway to provide for parking space and a corridor where materials may be transported between the road and the cottage. The *Cottage Architecture* might specify that the driveway should connect with the road. But what occurs beyond the end of the driveway is beyond the scope of the architecture, and as such, the roadway can be represented in the *Cottage Architecture* as a *Terminator*; something that exists and interfaces with the parts of the architecture, but is beyond the scope of the architecture.

Another good example of a *Terminator* is a power utility. The *Cottage Architecture* need not describe how or where electricity is generated, only where it enters the lot through the hydro lines. A *Terminator* may be used to represent the connection with the hydro lines that exist beyond the cottage property.

The *Logical* and *Physical Architectures* both have exactly the same set of *Terminators*.

Market Packages

As was noted in the beginning, a cottage means different things to difference people, and while the *Cottage Architecture* might address an exhaustive list of *User Services*, the cottagers might only be interested in a select few. A *Market Package* is a grouping of specific processes of the *Logical Architecture* and pieces of the *Physical Architecture* required to provide a defined set of *User Services*. *Market Packages* provide an accessible, deployment-oriented perspective to the architecture.

For example, in the *Cottage Architecture*, the *Watersports Market Package* might prescribe purchasing a storage shed, a basic dock, a boat, and short list of other assorted equipment to provide for the *Boating User Service*, the *Waterskiing User Service*, and the *Fishing User Service*. On the other hand, the *Comprehensive Waterfront Market Package* might prescribe a full-sized dock instead of a basic dock. This *Market Package* might offer the additional benefit of the *Sunbathing-On-The-Dock User Service*.

Canada's *ITS Architecture* includes a number of *Market Packages*, for example, the *Regional Traffic Control Market Package*, which provides the *Regional Traffic Control User Sub-Service*. It requires the *Roadway Subsystem* and *Traffic Management Subsystem* of the *Physical Architecture*, and includes a number of process specifications from the *Logical Architecture*.

Notably, as in the example above, there are a number of *Market Packages* that map directly to a single *User Service*. However, there are a number that map to more than one *User Service*. Furthermore, because *Market Packages* are evaluated during the architecture development, supporting benefits and costs analyses are also available for the *Market Packages*.

Time to Hit the Road *(But Leave the Waterskis at Home)*

That, in a nutshell, is what an *ITS Architecture* is, and what its components are.

Using an *Architecture* can help alleviate many of the problems that my father-in-law has encountered at his cottage:

- Assets being misused to address overlooked needs;
- Ad-hoc additions that compromise the effectiveness of the original system;
- Poor use of resources;
- Duplication of processes and unnecessary redundancy; and primarily
- Inefficiency and waste.

The intelligent use of *ITS Architectures* offers the potential to improve the transportation network, and if one thing is to be taken from this document, remember that the whole cottage thing, was just an analogy.

Don't include a *Boating User Service* in your *ITS Architecture*!!