

Life (yet to be) Scripted

Bits Versus Electrons

Life (Yet to Be) Scripted

By Bob Frankston

The fictional character Jeeves has come to represent the attentive servant doing our bidding. Unlike Jeeves, computers are mindless; they follow our instructions literally.

This literalness allows us to create very complex programs out of many thousands and even millions of individual instructions. Taking this idea of programming (or scripting) to the world of devices and infrastructure is something else.

When we talk about smart homes and smart cities, we are expecting computers to act like Jeeves, and we wind up frustrated when they do not do what we want. Perhaps worse, we become prisoners of our own device.

AUTOMATONS?

The idea of having servants who can do our bidding is ancient and, indeed, much of our excitement about home control taps into the idea of having a butler catering to every whim and even sometimes anticipating our wishes.

Our wonderful computing engines and devices are simply the fulfillment of this dream. We can now have machines that do our bidding. Just like our butlers understood us, our smart devices and smart cities will serve us very well. Yes, in having such expectations, we set ourselves up for disappointment.

It's easy to see the value of a rule such as "at sunset, turn on the porch light." However, that rule is not as simple as it appears on the surface. There is a lot of

implicit knowledge here, such as where we are and how we determine when sunset is. For that matter, do we really mean sunset as per the almanac or just "when it gets dark."

JEEVES IN CHARGE

Jeeves, the butler, would know what we really mean and might turn on the porch light early on a cloudy day and maybe later on a bright moonlit night. You could also casually tell Jeeves that you want to leave the light off because you want to look at the stars, or perhaps Jeeves will simply see that you are looking at the sky and choose to leave the light off.

Unlike Jeeves, computers are mindless; they follow our instructions literally.

The rules for the light seem just like another computer program and, to a degree, they are. The difference is one of context or attitude. We consider a program to work if it meets a set of well-defined criteria. Once we define the parameters, such as when sunset is and what message we should send to the light source, we consider the program to be done. And it is ... for the moment.

That is very far from saying that the program did what it was intended to do in the real world, in which context and wants matter. Asking why the chicken crossed the road is funny because children understand the dissonance between the words used and the deeper meaning.

Yet I'm often told that a program or product is correct because it performs according to a specification.

JEEVES AS A SERVICE

One way of adding value is to provide Jeeves as a service in standalone devices and online. The idea is that by accumulating enough knowledge about you, the service can add value to products or services in their own right.

A product like Siri can be remarkable in what it can do for you, but it can also fail spectacularly when we start to expect it to really understand what we want when we might not even know ourselves. Of course, Jeeves is not omniscient, and we may not really want that extra helping of dessert.

One reason why smart services work so well is that they can take advantage of the information available in our calendars, address books, and even our search history. And when they are right, it is very convenient. However, just because I looked at the score of one baseball game to see if the game was over yet does not mean I want Google to start showing me the scores for future games by that team.

"Big data" can let us see what is not obvious. To understand this, think of walking past a fence. It may seem opaque when you are standing still, but as you run by, your brain integrates images between the slats giving you a view of the yard as if the slats were transparent.

However, the algorithms used by computers do not necessarily correspond to how I understand the world, so it is

Digital Object Identifier 10.1109/MCE.2013.2280941
Date of publication: 18 December 2013

JANUARY 2014 IEEE CONSUMER ELECTRONICS MAGAZINE 57

If you can you should read the [IEEE Version](#) of this article where it is better formatted and also provides the IEEE with feedback about readership.

Note that my previous article is "[Deconstructing "the Smartphone"](#)".

The fictional character Jeeves has come to represent the attentive servant doing our bidding. Unlike Jeeves, computers are mindless. They follow our instructions literally.

This literalness allows us to create very complex programs out of many thousands and even millions of individual instructions. Taking this idea of programming (or scripting) to the world of devices and infrastructure is something else.

When we talk about smart homes and smart cities we are expecting computers to act like Jeeves and we wind up frustrated when they don't do what we want. Perhaps worse, we become prisoners of our own device.

Automatons?

The idea of having servants who can do your bidding is ancient and, indeed, much of our excitement about home control taps into the idea of having a butler catering to every whim and even sometimes anticipating our wishes.

On the surface our wonderful computing engines and devices are simply the fulfillment of this dream. We can now have machines that do our bidding. Just like our butlers understood us our smart devices and smart cities will serve us very well. Yes in having such expectations we set ourselves up for disappointment.

It's easy to see the value of a rule such as "at sunset turn on the porch light". But that rule isn't as simple as it appears on the surface. There is a lot of implicit knowledge here such as where we are and how we determine when sunset is. For that matter do we really mean sunset as per the almanac or just "when it gets dark".

Jeeves in Charge

Jeeves, the butler would know what we really mean and might turn on the porch light early on a cloudy day and maybe later on a bright moonlit night. You can also casually tell Jeeves that you want to please leave the light off because you want to look at the stars or perhaps, Jeeves will simply see you are looking at the sky and choose to leave the light off.

On the surface the rules for the light seem just like another computer program and, to a degree, they are. The difference is one of context or attitude. We consider a program to work if it meets a set of well-defined criteria. Once we define the parameters such as when sunset is and what message we should send the light source we consider the program to be done. And it is ... for the moment.

That is very far from saying that the program did what it was intended to do in the real world in which context and wants matter. Asking why the chicken crossed the road is funny because children understand the dissonance between the words used and the deeper meaning. Yet I'm often told that a program or product is correct because it performs according to a specification.

Jeeves as a Service

One way of adding value is to provide Jeeves as a service in standalone devices and online. The idea is that by accumulating enough knowledge about you the service can add value to products or services in their own right.

A product like Siri can be remarkable in what it can do for you but it can also fail spectacularly when we start to expect it to really understand what we want when we might not even know ourselves. Of course Jeeves isn't omniscient and we may not really want that extra helping of dessert.

One reason why smart services can work so well is that they can take advantage of the information available in our calendars, address books and even our search history. And when they are right it is very convenient. But just because I looked at the score of one baseball game to see if the game as over yet doesn't mean I want Google to start showing me the scores for future games by that team.

"Big data" can let us see what is not obvious. To understand this, think of walking past a fence. It may seem opaque when you are standing still, but as you run by, your brain integrates images between the slats giving you a view of the yard as if the slates became transparent.

But the algorithms used by computers don't necessarily correspond to how I understand the world so it's difficult to be explicit about what I want. I must trust Jeeves.

Techniques like building in fancy rules and using the data around us allow us to create many useful service and they demo well. But the real world is ambiguous and not hierarchical and thus these techniques don't scale well beyond their design point. We might define a table as a surface sitting on legs but we don't care about the dictionary definition when we just need to put down a glass of water.

Keeping things Simple

Electronic spreadsheets represent a different approach. Instead of acting as Jeeves they give your inner Jeeves a chance to perform by providing a blank sheet that can perform the mechanics of calculating and, more to the point, recalculating numbers when the parameters change.

Spreadsheets work well when we have a sense of the relationships between the numbers and see how information in one cell affects information another cell. But we cede control and run a risk when we expect magic and forget that the numbers themselves may not be meaningful.

That said they do give people what is, in effect, a tool that allows them to script the task of performing repetitive calculations.

I want what I want

The neat trick of simple spreadsheets is that you build them as part of creating a sample of what you want. Simple web pages are similar – you add descriptive markings like "header" or "bold" and can see the result.

This is also why Jeeves is so helpful – you just know you want the porch light on and don't think about how it happens.

With traditional programming you need to specify each step in excruciating detail and slowly build up layers of abstraction to the point that you can send a "turn on porch-light" message instead of worrying about the exact mechanics of sending power to a light bulb so it shines.

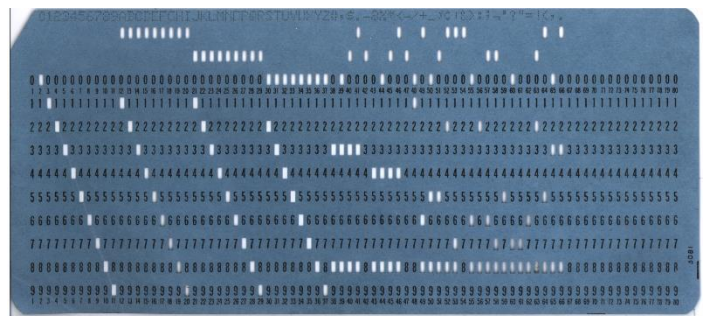
Things get a little more complicated when we are dealing with events and time. Do you really want to say "turn the light on at sunset (such as 7PM)" or do you mean to say "assure the light is on between sunset and sunrise"?

No wonder even after all this time we haven't replaced Jeeves. There's a slow process of gaining experience and developing best practices for this new world of programming things. Not just programmable but connected.

Even something as simple as a porch light becomes very complicated when we try to have rules interact. What if family members have different rules that don't mesh? In the real world we somehow work it out. It's much harder with rules that are just mindless and can't anticipate situations we haven't thought about.

Making Connections

I started my career as computing was making the transition from standalone computing in which a program would read a single set of data and produce a result. Today few people could even imagine a time when the primary storage medium was a piece of cardboard with holes punched in it to represent the data – those chads made infamous in the 2000 election.



Today we are constantly connected and create connections among devices and interconnect information. Much of the information we use is gathered in databases and then we have programs that operate on them and, perhaps, display a web page and give us the ability click on icons to buy products.

Some of us write code to parse the HTML to get at the rich lode of data hidden within – sort of like cracking open a

walnut. But it only hints at what is possible. Without a well-defined interface any such code is brittle. And not all the information is even available – we just see a picture of what we want. There is also a worrisome trend to force us back into manually interacting with the web sites because the empowered user is seen as a threatening hacker rather than a creative prosumer.

What's the Incentive?

Why would a company selling smart devices for my house want go through the trouble of supporting users who are going to blame them for bugs in their codes? Today we are also becoming aware of the challenge of protecting against malicious snippets of code – what we call viruses and malware.

Is this dream of a scriptable world in which we could add intelligence everywhere going to fade away like the future in which we'd all use auto-gyros to get to work?

If you look at the ads from companies selling us smart cities and smart devices you'd wonder why we'd want to bother our pretty little heads with programming our world when Jeeves will take care of it for us.

But the Jeeves mines the past. Tools that give us control, like spreadsheets, give us the ability to discover new possibilities.

Looking at the excitement on Kickstarter when people offer devices that start to give us the ability to control devices even ones as simple as light bulbs there is indeed a real demand for such capabilities. But before we can realize this future we first need to understand the challenges we face both technically and in terms of business.

For now, I will continue to program around these limits as we develop the techniques that can indeed give us the ability to share control of the world around us and add little dabs of intelligence without relying on Jeeves' omniscience.