

“Promise Theory for Dummies” @botchagalupe

“History of Management” @botchagalupe

Friday, February 22, 13

Gonna take a little diversion first..

What I found when researching this presentation was there was a much more interesting story here.

John Willis
@botchagalupe
Enstratus
VP of Customer Services



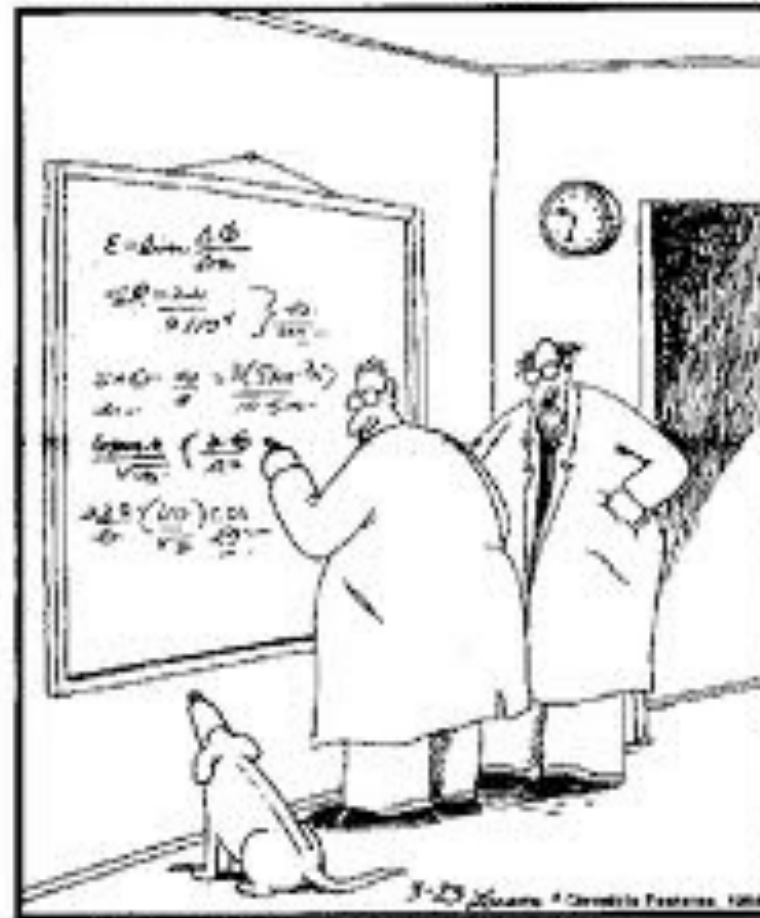
Friday, February 22, 13

We do cloud management and of course I am hiring (like everyone else)
Anything more regarding that please come talk to me...

- ▼ **Deterministic vs Indeterministic**
- ▼ **Obligation vs Cooperation**
- ▼ **Exact vs Inexact Science**
- ▼ **Classic vs Quantum**
- ▼ **Absolutes vs Probabilities**

THE FAR SIDE

By GARY LARSON



"Ohhhhhhh ... Look at that, Schuster ...
Dogs are so cute when they try to comprehend
quantum mechanics."

Friday, February 22, 13

I want to start you thinking about these words and their differences as a placeholder for the rest of this discussion.

Indeterminism is the concept that events (certain events, or events of certain types) are not caused, or not caused deterministically (cf. causality) by prior events. It is the opposite of determinism and related to chance.

Correlation and causation are basically the same vs Correlation is not Causation.

How I got here - 2011?

- ▼ **Mark Burgess**
- ▼ **Gene Kim**
- ▼ **Ben Rockwood**



5

Friday, February 22, 13

I was giving a presentation here 2011 and Chef & Puppet - mark story...
Also in 2011 I met Gene Kim (Founder of tripwire, visible ops, the Phoenix project (Goldratt/TOC) , devops cookbook
@benr gave an awesome 2011 LISA keynote on devops transformation... and (deming)

How I got here - 2012?

Join us for PuppetConf 2012

September 27th & 28th at the Mission Bay Conference Center, San Francisco

PuppetConf Preview: "Putting the Science Back From Deming and Goldratt to DevOps"

Posted on September 14, 2012 by Michelle Lopez

John Willis from Enstratus

Analyzing a Complex Cloud Outage



@botchagalupe

VP of Services enStratus

IT REVOLUTION PRESS

HELPING SPARK THE IT CAMBRIAN EXPLOSION

BLOG

[Deming to DevOps \(Part 1\)](#)

By John Willis

Edward Deming is often referred to as the father of quality and I

[Deming to Devops - DevOpsDays](#)

devopsdays.org/events/2012-newyork/.../

Deming to Devops is a journey through the thoughts of thought leaders and their influence on modern DevOps.

Friday, February 22, 13

Spent most of last year researching deming and his impact on devops...
Using System of Profound knowledge.. on amazon outages...

- ▼ **History of Management**
- ▼ **History of Config Management**
- ▼ **Promise Theory**



Friday, February 22, 13

--Enough foreplay...

--This is an outline of this presentation.

-- History of Man is important for 2 reasons. 1) gives us a better idea how we get to PT and 2) also explains how 2 devops

--Walk through what I am calling the history of management. (it may be a boodoggle but I hope it's interesting and u find it related)

--Since CM is near and dear to my heart I think we can use CM as a base line for this discussion related to in/determ topics expressed earlier

--Use complex systems as an anchor to explain promise theory...

History of Management Timeline - Pre 20th Century

1000000 BC
Stone Tools

8000 BC
Fitted Parts

1687 AD
**Principia
Mathematica**

1787 AD
**Interchangeable
Parts/Division of
Labor**

1840 AD
go-no-go tolerance

Deterministic

Indeterministic

1859 AD
Origin of Species

1872 AD
Statistical Mechanics

1890 AD
Chaos Theory

5

Friday, February 22, 13

Shewhart's book timeline "Statistical Methods" .. history of quality control...

Newtonian Physics.. Fact based thinking deterministic thinking.

Ilya Prigogine "The end of Certainty"

Ludwig Eduard Boltzmann - Statistical Mechanics -

He tried to emulate what Darwin had done in biology as an evolutionary approach to physics (multip particles (100 sextillion) 21 zeros...

Henri Poincaré on Chaos Theory..

History of Management Timeline - 20th Century

1911 AD
Principles of Scientific Management

1927 AD
Mass Production

1950 AD
Post War Neo Taylorism

1916 AD
Relativity: The Special and General Theory

1940 AD
Decentralized Management Alfred P Sloan

Deterministic

Indeterministic

1924 AD
Control Charts

1950 AD
Deming in Japan

1988 AD
Lean Manufacturing

1984 AD
The GOAL (TOC)

1927 AD
Uncertainty Principle

1956 AD
Toyota Production Systems

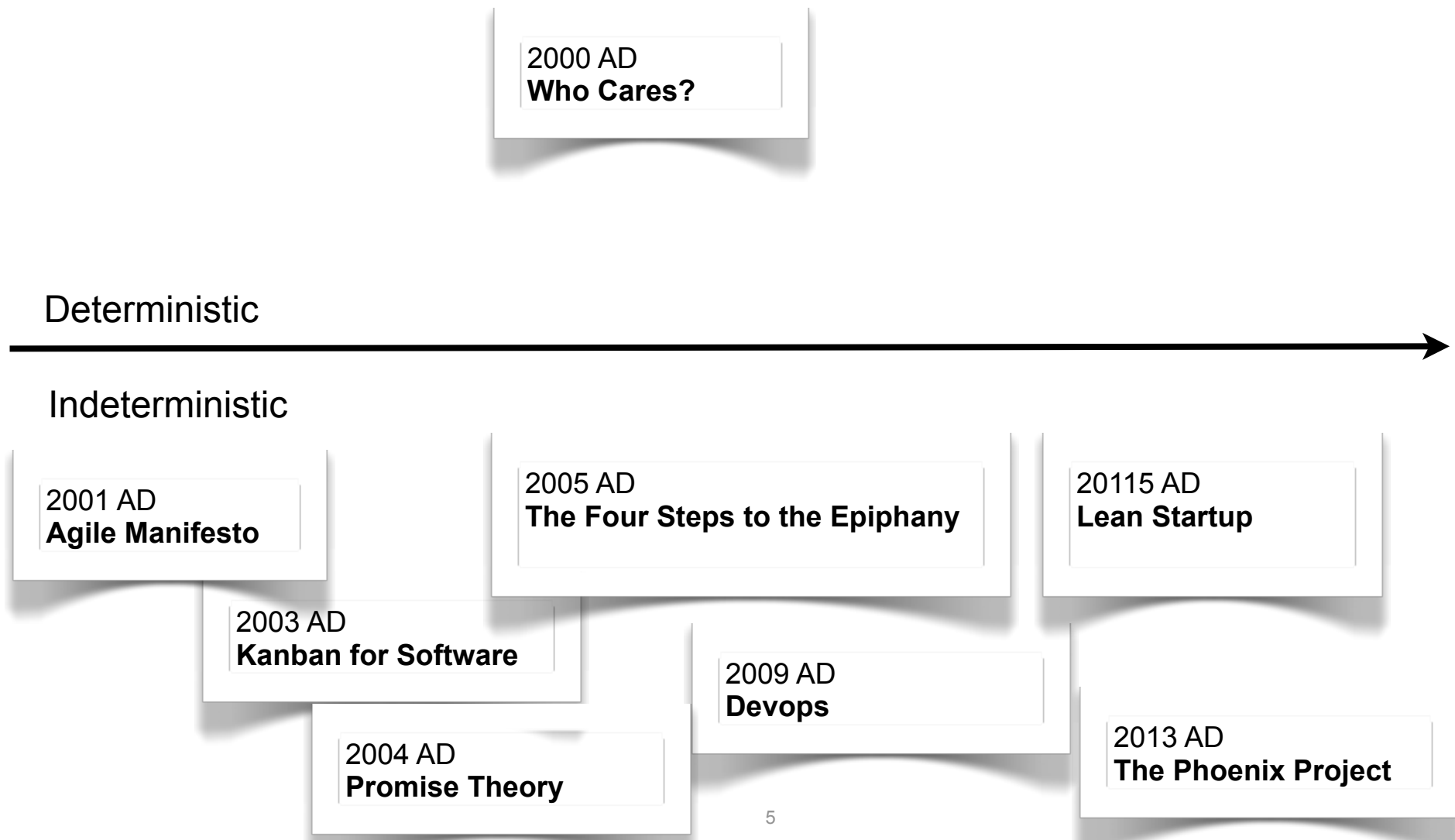
1993 AD
System of Profound Knowledge (SoPK)

5

Friday, February 22, 13

- Lean coined by John Krafcik in his 1988 article, "Triumph of the Lean Production System,
- known to the world through Jim Womack's The Machine That Changed the World
- Important point here... Shewhart, Deming, Goldratt were all Physicists ..
- 20-th century was a sort of "end of Certainty... way of thinking... that not only influenced physics but influenced management science...

History of Management Timeline - 21st Century



Friday, February 22, 13

Two important themes,,,

- 1) the mapping of events that directly relate to what we call devops and promise theory
- 2) Turn of the 20th century there was the new renaissance thinking rooted in indeterminism..

15Mins

History of Configuration Management

- ▼ **First Generation**
 - ▼ Tivoli, HP, BMC
- ▼ **Second Generation**
 - ▼ CFEngine, Puppet, Chef
- ▼ **Third Generation**
 - ▼ Complex Systems



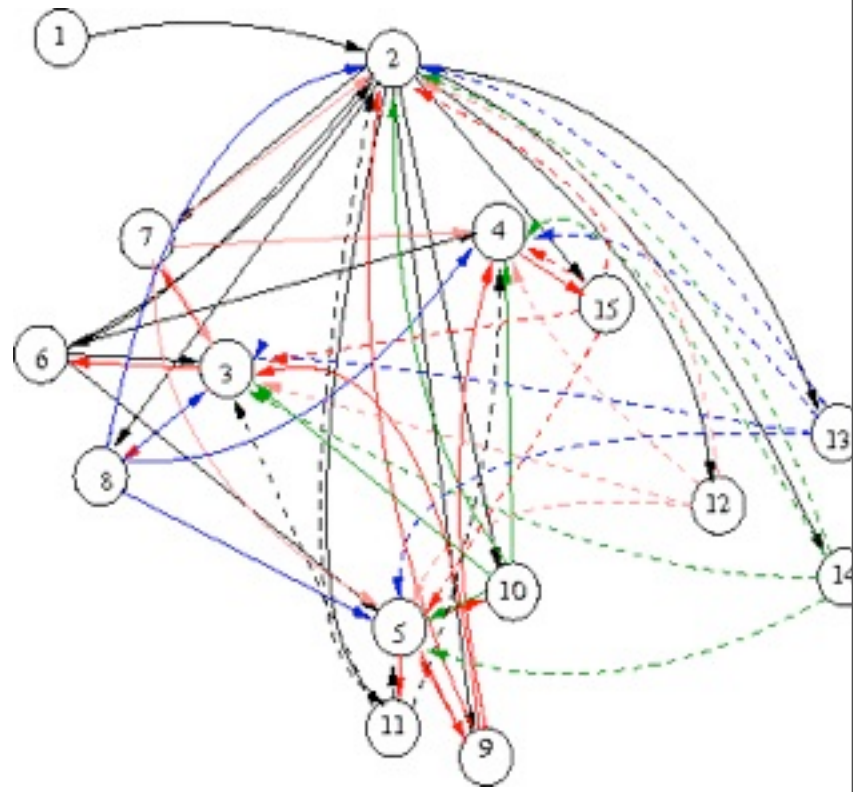
Third generation needs to be based on complex systems (indeterministic models, probability based models)

-- It's only logical to contend... As the rate of change increases, scale increases and complexity increase so does IT Entropy...

-- Tell the scale deterministic vs in deterministic model..

Promise Theory

- ▶ **Promise theory is a model of voluntary cooperation between individual, autonomous actors or agents**
- ▶ **Who publish their intentions to one another in the form of promises.**



Implementations of Promise Theory

- ▼ **Economics and Game Theory**
- ▼ **Organizational Behavior**
- ▼ **Systems Management**
- ▼ **Quantified Self Movement**

Patterns of Obligations vs Cooperation

▼ Failure

- ▼ It must not fail

- ▼ It will fail

▼ Automation

- ▼ It must look like this

- ▼ It should look like this

▼ Visualization

- ▼ Show me what it looks like

- ▼ Show me what you think it looks like



Obligations vs Promises

- ▶ **Obligation is not a promise**
- ▶ **Obligations are a problematic ideal**
- ▶ **Obligations oversimplify information**
- ▶ **Obligation allows you to take things for granted**
- ▶ **Obligations can come from anywhere**

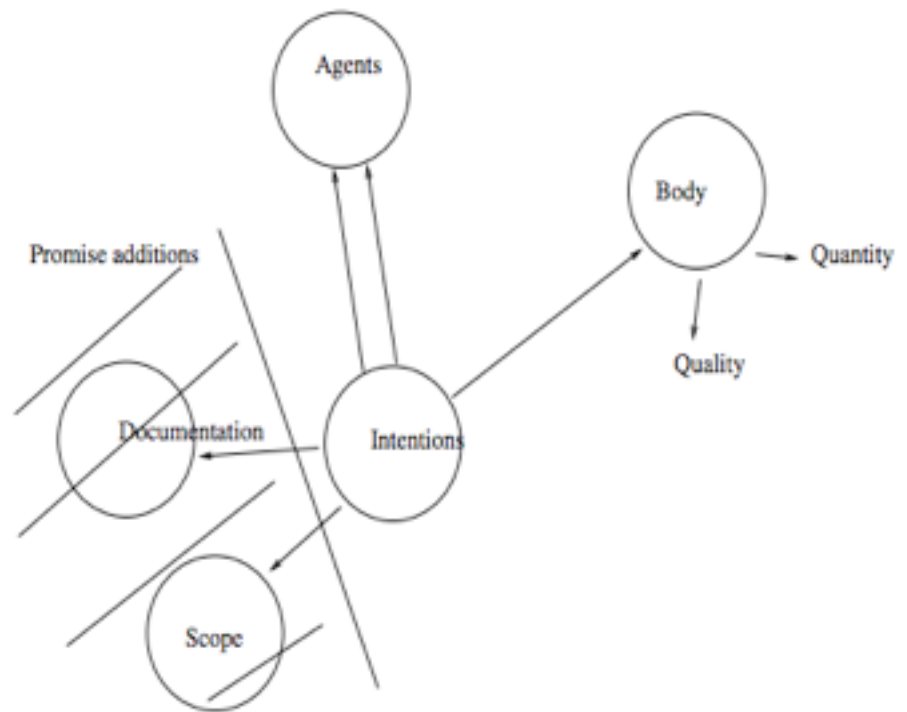
5

Friday, February 22, 13

- An obligation is deterministic and a promise is indeterministic
- An obligation assumes facts...
- Over simplification in that you know all the external agents reactions before hand.
(You will look like this)
- You have simplified all the edge cases .. DNS must work...
- Tell the devopsworkshop story... large telco ... never, never, never break... never never never
- Airplanes break all the time.. it's not about them not breaking it's about how you handle the breaks...
- Apollo 13 Ed Harris.. great failure story..
- promises impy contingency.. obligations do not..
- A promise is a I'll do what i can based on what I know as opposed to this must happen even though the ..
reality is there are a lot of things out of my control...

Deconstructing Promises

- ▶ There must be agents in order for promises to exist
- ▶ Intention is a description of behavior
- ▶ Body explains the quality and quantity of the intention
- ▶ Quality is (type, topic or subject for the promise)
- ▶ Quantity (how much of the realm of possibility for that subject is being promised)



5

Friday, February 22, 13

--Key point intention becomes a promise once it is a) documented and b) it has scope (exposed to another agent)

promise quality:

“I will feed your cat”

promise quantity:

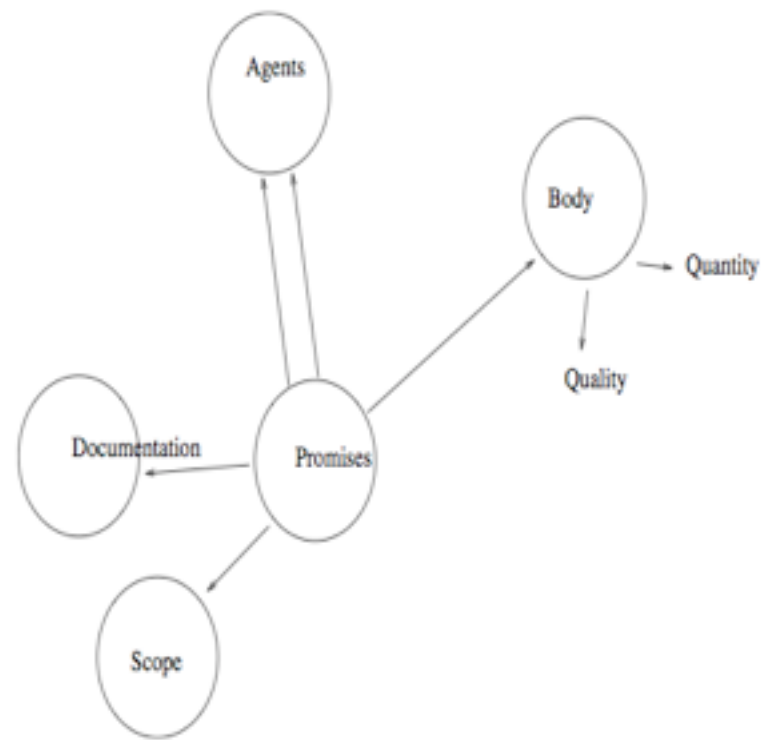
“on Monday &
Friday mornings”

promisee:

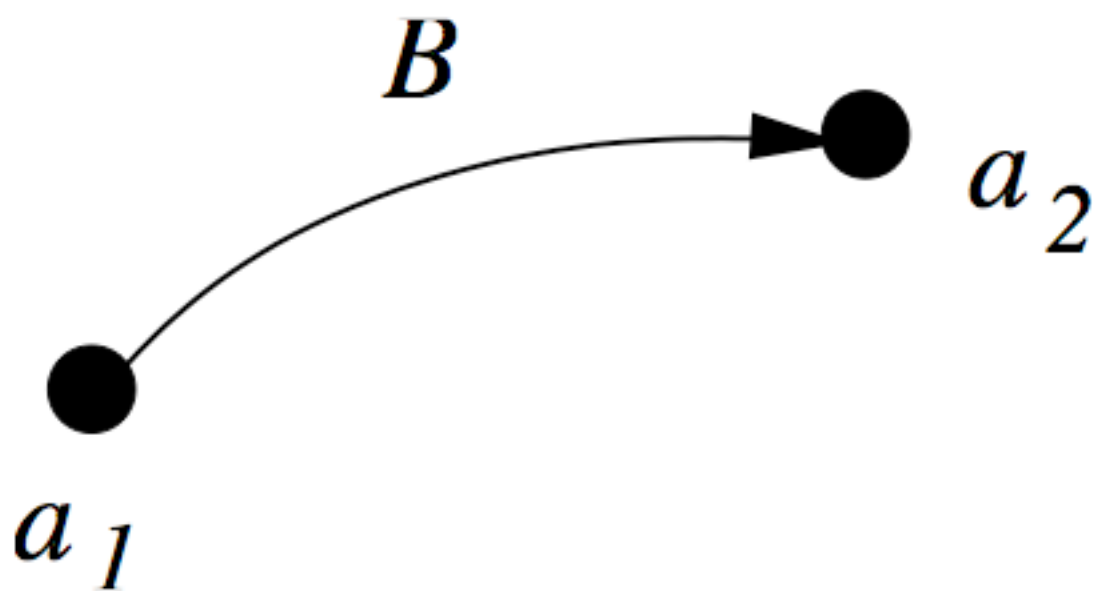
“my neighbor”

promisor:

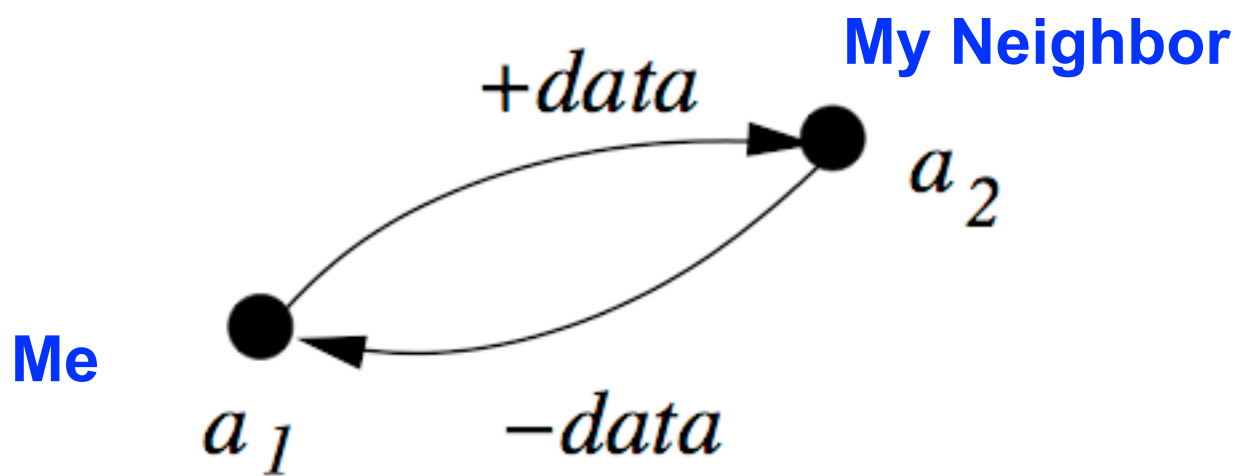
“me”



Deconstructing Promises



Deconstructing Promises



□

5

Friday, February 22, 13

There are promises to give and promises to receive

We use + for give and - for receive..

Let's use the feed the cat example...

I promise to feed the cat (a_1) and (a_2) promises me access to get in his house...

This leads to the concept of trust related to promise theory...

My neighbor makes the assumption that I am reliable and won't trash his house drink his 25yr old scotch.

I make the assumption that I can get in his house (he gave me the correct alarm code).

For example I don't want to have to wake up early to find out I can't get in or even worse the alarm goes off (get arrested)

-This also ties in an economic relationship.. maybe my N makes more money on M & F and/or he pays me.

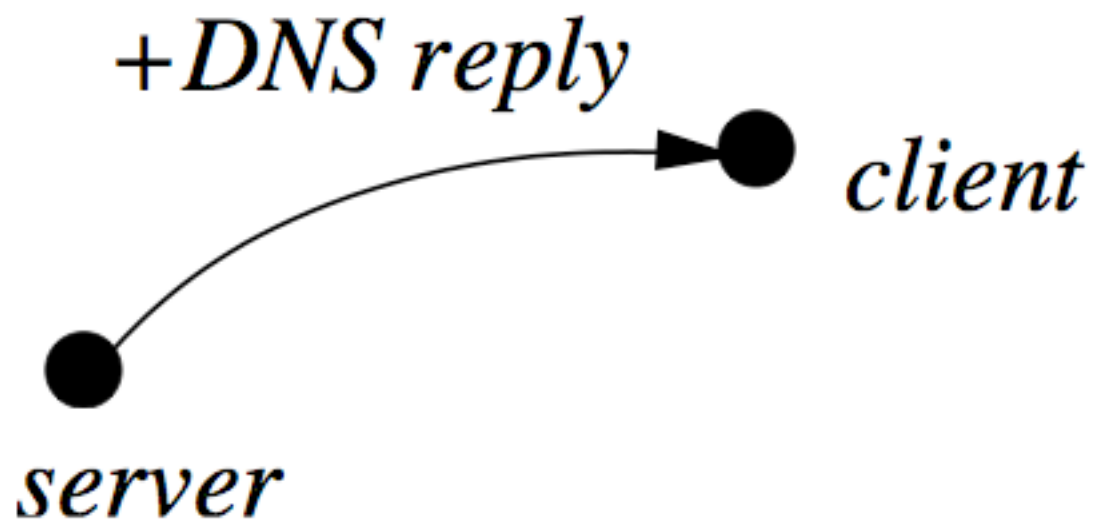
-Maybe I do it to be a good neighbor or I know I will need his help in the future...

You see how this voluntary cooperation can start to manifest...

Last but by no means least, because promises anchor a sender to a receiver with a directional arrow, they form networks and hence there are two kinds of promises: promises to give and to receive. We write these using a + (give) and a - (receive) sign.

Here, the source of the data a_1 promises to make the data available to a_2 (perhaps by running an appropriate daemon and granting some access controls (for a cfengine-like

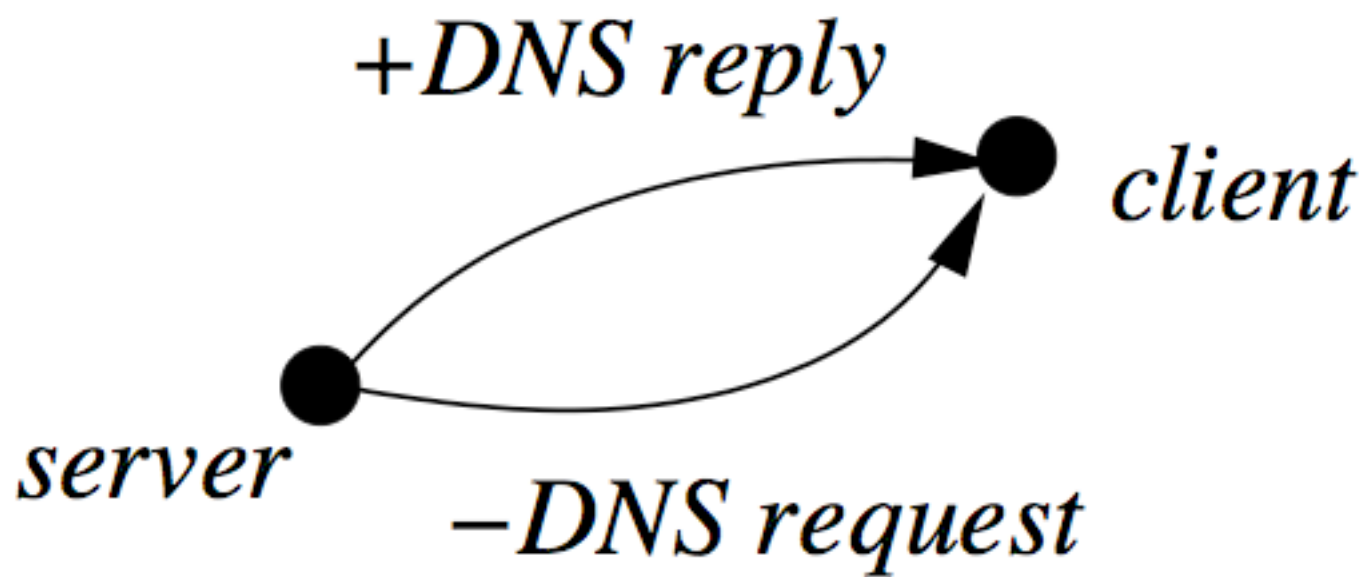
DNS Example



Friday, February 22, 13

- Let's see how promise theory helps us to understand something we think is obvious: the Domain Name Service.
- Given that promises are service oriented, we might think that it is trivial to model this service by a single promise from a server to each client.
- The promise made by the DNS server to reply to requests is certainly important:

DNS Example

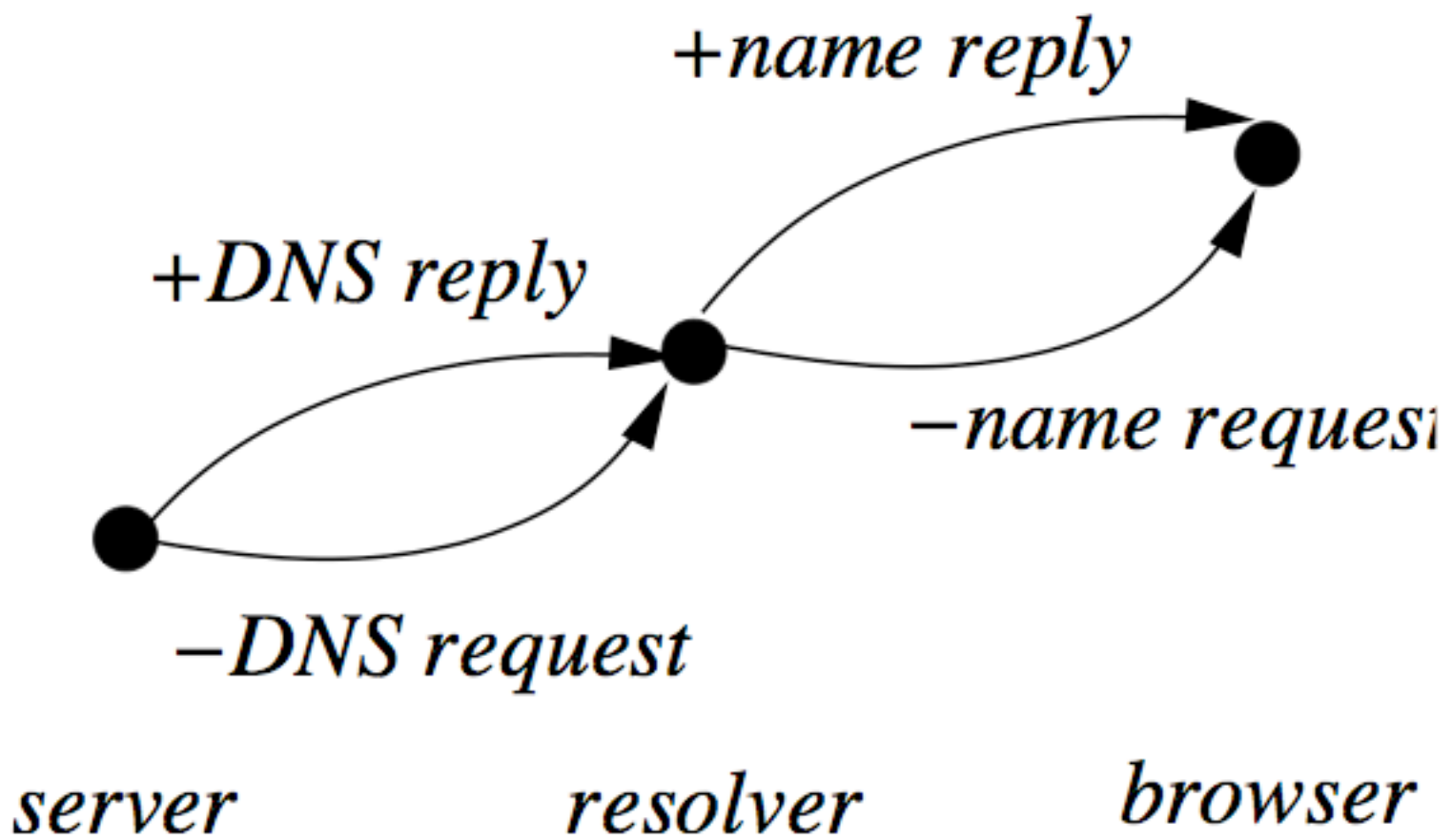


5

Friday, February 22, 13

- Actually the DNS service makes a general promise to provide query information.
- This, in fact, involves two promises
- The server promises receive requests (-)
- and it also promises to give a reply (+)
- Important note here is that the agents are autonomous.

DNS Example

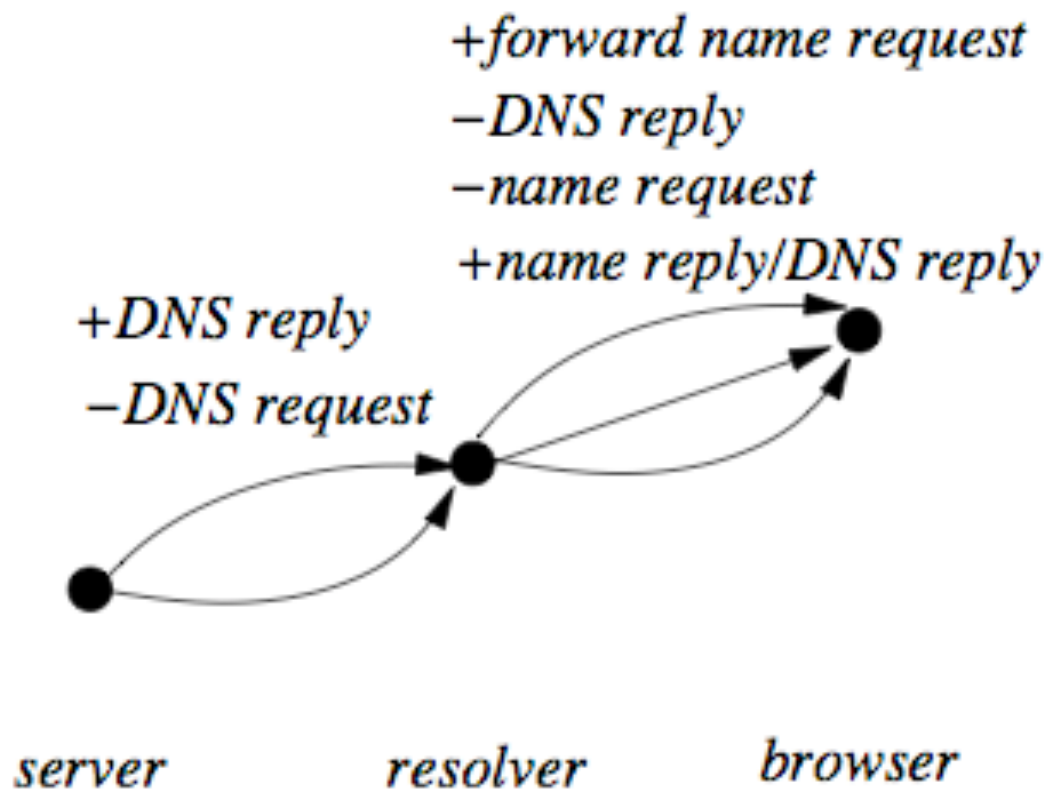


5

Friday, February 22, 13

- Lets look at a more complicated view of this relationship... where there is a browser, resolver and server...
- The local resolver promises to give some answer to the web browser and to accept its requests.
- Key point here is the resolver has only agreed to accept requests and return request. It has not agreed to forward the request in it's promise to the browser.

DNS Example



5

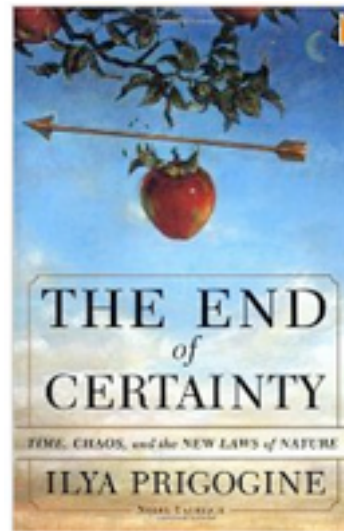
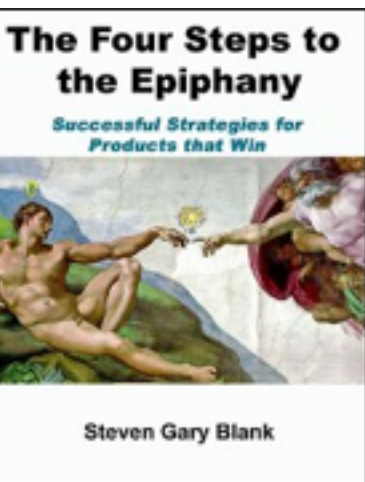
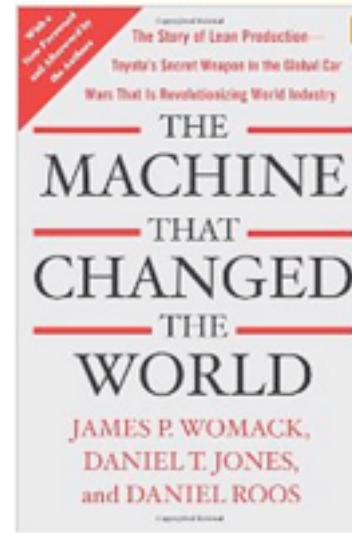
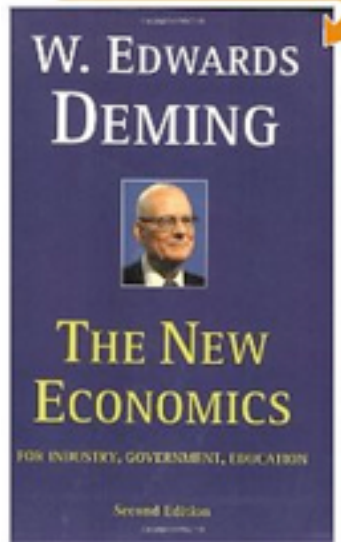
Friday, February 22, 13

- We need promises from the resolver
- It will forward the request and it receive the request
- Can you see how this level of decomposition and atomic thinking start taking a bunch of assumptions out of the picture...
- Forces you to stop taking things for granted...
- Bring back the point of complexity and CAS... in Avionics there isn;t one big box that says all things are working or not..
- It's lots of little autonomous agents cooperating to react and correct failures...
- important point.. including the human..
- The human becomes an agent as part of the CAS not the engine that drives the model

Key Points

- **One of the strengths of this way of thinking is that it does not let you take things for granted.**
- **Decomposing into agents and promises allows us to see how a system works.**
- **Confronting detailed (atomic) expectation of behavior, not just the ones that we think are big and important.**
- **The entire chain of responsibility begins to appear as we start thinking in these terms.**

Putting the Science back into IT



Friday, February 22, 13

All of those guys have read these books....

What do you think these guys have in common? All engineers two started out as physicists...

Reference Material

- ▼ <http://research.iu.hio.no/promises.php>
- ▼ http://cfengine.com/markburgess/blog_devops.html
- ▼ <http://www.amazon.com/Total-Quality-Information-Systems-Technology/dp/1884015700>
- ▼ <http://www.youtube.com/watch?v=4CCXs4Om5pY>
- ▼ <http://www.slideshare.net/jurquhart/gluecon-keynote>
- ▼ <http://www.kitchensoap.com/>