

The importance and potential of descriptions to our industry

Professor Steve Ross-Talbot
June 2015
V1.0

steve.ross-talbot@cognizant.com
www.cognizantzdlc.com



Overview

- **Descriptions**
- **Why they are important**
- **How we do it today in industry**
- **What we can do with good descriptions (potential)**
- **How we get there (the legacy issue)**

Descriptions

Kate Middleton Proves Once Again That She's (Sort Of) Like Us

Happy Friday! Here's every song that sounds like 'Uptown Funk'

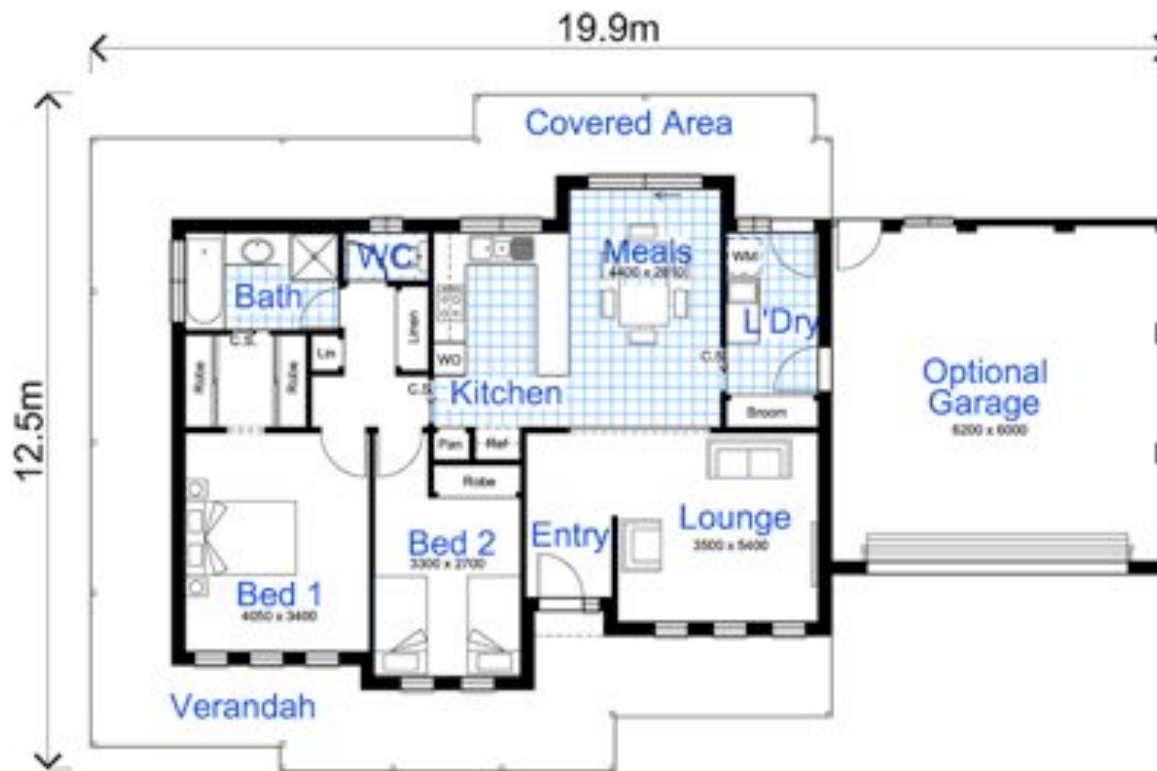
Camembert de Normandy smells like the secret project of a chemical company.

I've got the moves like Jagger

That IT system is like a knotted ball of wool. Spaghetti plate of.

Build me a house like:

Descriptions



Descriptions and Validity

"The present King of France is not bald"
Bertrand Russell

Just because we describe it doesn't mean it is valid

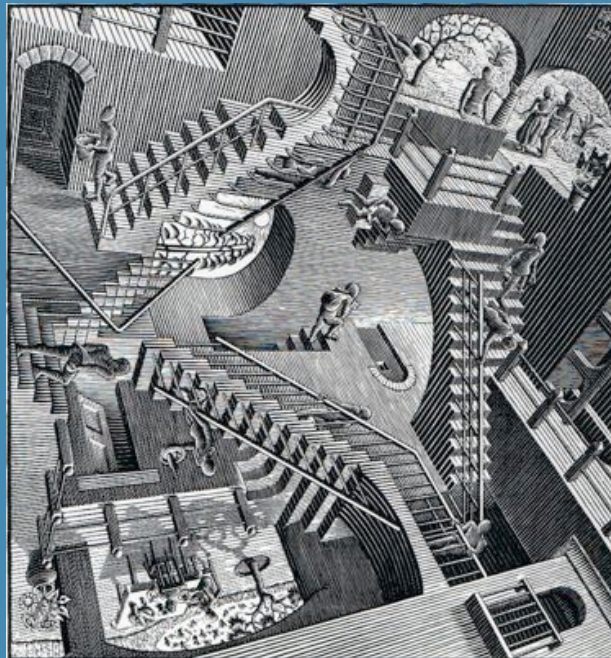
$$\forall x[\text{PKoF}(x) \rightarrow B(x)]$$

$$\exists x[\text{PKoF}(x) \ \& \ \forall y[\text{PKoF}(y) \rightarrow y=x] \ \& \ B(x)]$$

Descriptions and Ambiguity

The importance of being Ernest

I need some stairs like:



Descriptions and Ambiguity

- **Ambiguity is the killer in our communication**
- **The cost of software defect to the US economy alone is expected to exceed \$85bn this year.**
- **Defects are caused by ambiguity and result in defect leakage between the phases of the SDLC and defect injection within those phases.**
- **Communication that is unambiguous is not just desirable it makes economic sense too.**

Ambiguity is both funny and dangerous

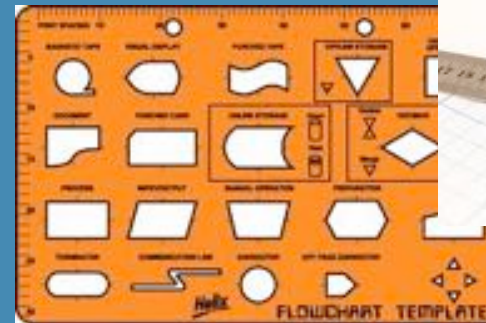
Play

Why do we describe things?

- **Requirements**
 - What is needed to meet some business objective
- **Models**
 - How the requirements will be met at some higher level
- **Applications/Code/Configuration**
 - How the requirements will be met at a detailed level
- **We describe so we can communicate effectively and efficiently**
 - We seek to communicate business objectives
 - We seek to communicate ideas
 - We seek to communicate how we intend to meet business objectives with ideas
- **If we had a magic brain hat that could take requirements and create solutions to them we would not need to communicate**

How do we describe things?

- We jot things down to explain things
 - We might use Word, eXcel and many other things.
 - We might use a RDBMS to capture and store and record relationships.
- We draw to explain things
 - We might use use cases to explain how something might work
 - We might use sequence diagrams to explain how something might work.
 - We might use UML or BPMN
 - We might use Visio



How do we describe things?

R0

We want to have a cross functional claims process available online for both agents and insured parties

Text – client verbatim

R1

No processing of a claim can occur until after a claim has been notified.
 Status enquiries and claims processing may happen in parallel.
 Only when claims processing has finished will a claim be settled.

R2

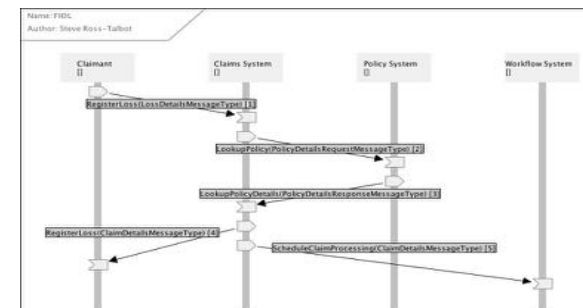
Policies will carry a Policy Reference
 Claims will carry a Claim Reference
 Loss Adjustors and claimants as well as agents will be provided with online access through a web channel.
 Existing systems will be reused as services.
 Existing services will be reused as is.
 A set of concrete use cases and supporting sequence diagrams.

Level 0 TO-BE



A typical jotting

Some suggested technical use case



Ref	Identity	Tokens	Identity Values	Query Expressions	Message File
[1]	null	null	null	null	LossDescriptionMessage.xml
[2]	null	null	null	null	PolicyDetailsRequestMessage.xml
[3]	null	null	null	null	PolicyDetailsMessage.xml
[4]	null	null	null	null	LossAcknowledgementMessage.xml
[5]	null	null	null	null	ScheduleClaimMessage.xml

How do we describe things?

R2

Policies will carry a Policy Reference

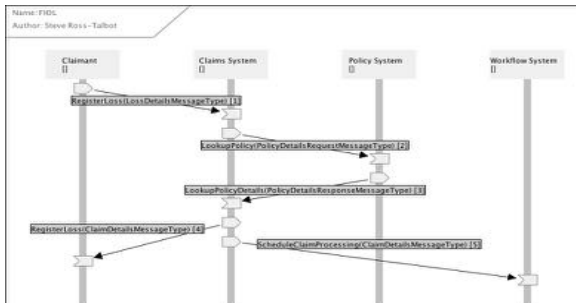
Claims will carry a Claim Reference

Loss Adjustors and claimants as well as agents will be provided with online access through a web channel.

Existing systems will be reused as services.

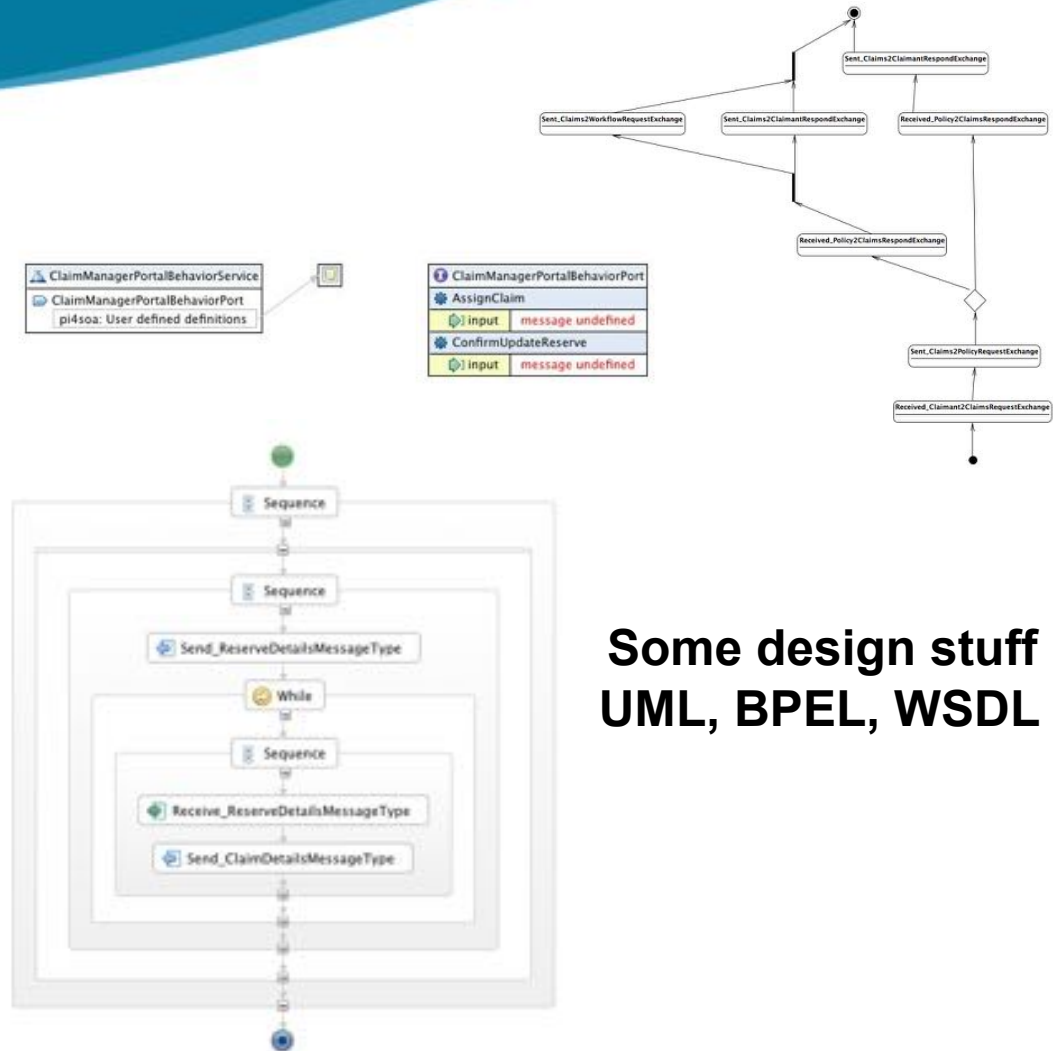
Existing services will be reused as is.

A set of concrete use cases and supporting sequence diagrams.



Ref	Identity	Tokens	Identity Values	Query Expressions	Message File
[1]	null	null	null	null	LossDescriptionMessage.xml
[2]	null	null	null	null	PolicyDetailsRequestMessage.xml
[3]	null	null	null	null	PolicyDetailsMessage.xml
[4]	null	null	null	null	LossAcknowledgementMessage.xml
[5]	null	null	null	null	ScheduleClaimMessage.xml

Some suggested technical use case



Some design stuff
UML, BPEL, WSDL

What are the problems in our descriptive forms?

- **Wrong conceptual model for the target audience**
- **No semantics, lack of formalism**
- **Which gives rise to a lack of hard core formal analysis and poor confidence**
- **TOGAF:**
 - A formal description of a system, or a detailed plan of the system at component level to guide its implementation
 - The structure of components, their inter-relationships, and the principles and guidelines governing their design and evolution over time.
- **With no formalism and wrong conceptual level we have difficulty in articulating and proving the impact and cost of change .**
- **Answering questions like how big and how much and even where to start become ad-hoc subject to magic and experience. And so not provable.**

What are the problems in our descriptive forms?

Answering questions like how big and how much and even where to start become ad-hoc subject to magic and experience. And so not provable.



© Scott Adams, Inc./Dist. by UFS, Inc.

What are the problems in our descriptive forms?

R2

Policies will carry a Policy Reference

Claims will carry a Claim Reference

Loss Adjustors and claimants as well as agents will be provided with online access through a web channel.

Existing system:

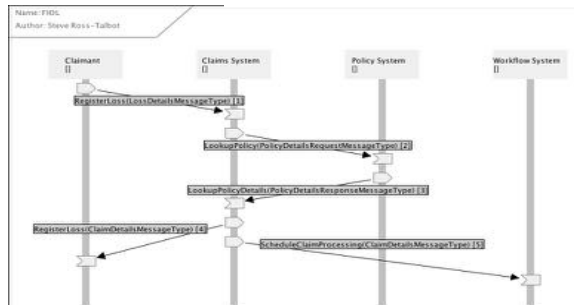
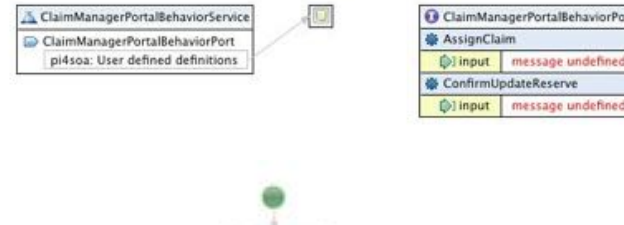
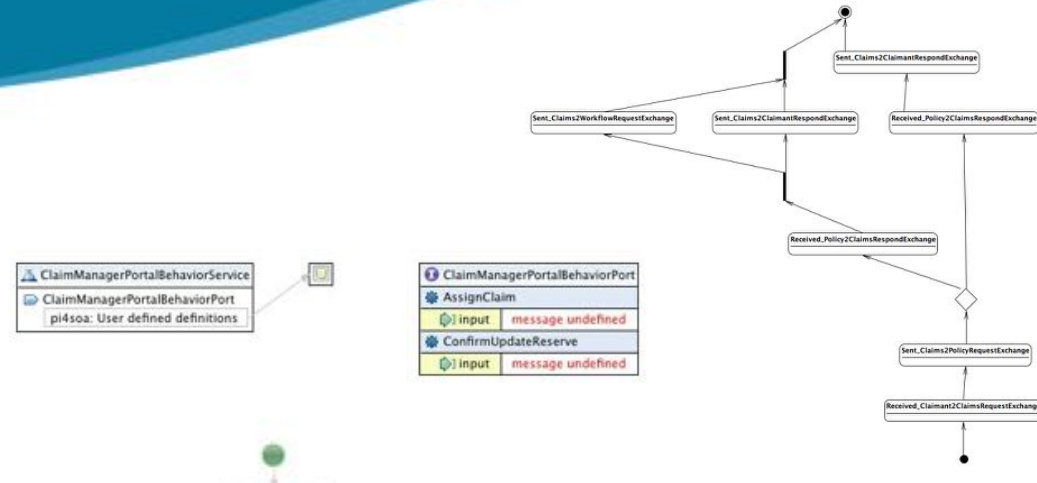
Existing service:

A set of concrete sequence diagram

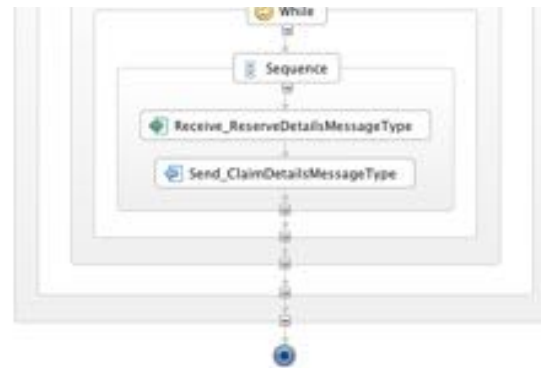
Where are the proofs of refinement?

Where are the proofs of abstraction? gn stuff

UML, BPEL, WSDL



Ref	Identity	Tokens	Identity Values	Query Expressions	Message File
[1]	null	null	null	null	LossDescriptionMessage.xml
[2]	null	null	null	null	PolicyDetailsRequestMessage.xml
[3]	null	null	null	null	PolicyDetailsMessage.xml
[4]	null	null	null	null	LossAcknowledgementMessage.xml
[5]	null	null	null	null	ScheduleClaimMessage.xml



Some suggested technical use case

How do we show the formal relationship between descriptions?

✓ TODAY ...

- ✓ Governance
 - ✓ Safety in numbers
 - ✓ Reviews and Sign-offs
- ✓ Yet another delivery method (Agile)
- ✓ Hope and pray and if I test enough and spend enough

✓ AND TOMORROW ...

- ✓ Proof Proof Proof
- ✓ Requirements are consistent, valid, etc
- ✓ Designs meet requirements
- ✓ Code meets designs
- ✓ Run time behaviors according to the designs
- ✓ Determine the scope or impact of a change
- ✓ Find the best behavioral match (something “like”)



$$\begin{aligned} \varphi &= \begin{pmatrix} E \\ H \end{pmatrix}, & E &= (\nabla \times H) \times \eta, & H &= \mu_0^{-1} \nabla \times A, \\ \text{and} & & & & & \\ & & A_x = A_y = 0, & & & \\ & & \begin{cases} \left(\cos \frac{\pi x}{l} \right)' \left(\cos \frac{\pi y}{l} \right)' \left(\cos \frac{\pi z}{l} \right)' \\ \text{for } |x| \leq \frac{l}{2}, |y| \leq \frac{l}{2}, |z| \leq \frac{l}{2} \\ 0 \text{ otherwise;} \end{cases} & & & \end{aligned}$$

How do we show the formal relationship between descriptions?

✓ TODAY ...

- ✓ Governance
 - ✓ Safety in numbers
 - ✓ Reviews and Sign-offs
- ✓ Yet another delivery method (Agile)
- ✓ Hope and pray and if I test enough and spend enough

✓ AND TOMORROW ...

- ✓ Requirements are consistent, valid, etc
- ✓ Designs meet requirements
- ✓ Code meets designs
- ✓ Run time behaviors according to the designs
- ✓ Determine the scope or impact of a change
- ✓ Find the best behavioral match (something “like”)
- ✓ Being able to say that this instance (run/trace) was compliant against it’s obligations (BIG BIG PROBLEM)



$$\begin{aligned} \varphi &= \begin{pmatrix} E \\ H \end{pmatrix}, & E &= (\nabla \times H) \times \eta, & H &= \mu_0^{-1} \nabla \times A, \\ \text{and} & & & & & \\ & & A_x &= A_y = 0, & & \\ & & \begin{cases} \left(\cos \frac{\pi x}{l} \right)' \left(\cos \frac{\pi y}{l} \right)' \left(\cos \frac{\pi z}{l} \right)' \\ \text{for } |x| \leq \frac{l}{2}, |y| \leq \frac{l}{2}, |z| \leq \frac{l}{2} \\ 0 \text{ otherwise;} \end{cases} & & & & \end{aligned}$$

Legacy



\$3.8 Trillion

\$3 Trillion Keep the Light on

Process Mining

Play

Process Mining

- **Listen to the Voice of the Machine**
 - Harvest the *session instance* from observations
- **Translate into AS-IS Model**
 - Generate the *session type* from the instances
- **Validate the AS-IS Model**
 - Test the *session instances* against the *session type*
- **Extend the AS-IS Model to incorporate the Voice of the Business into a TO-BE Model**
 - Incorporate new requirements, new *session instances*
- **Validate the TO-BE Model**
 - Test the *session instances* against the *session type*

Q&A

