

Applicability of SSM and UML for Designing a Search Application for the British Broadcasting Corporation (BBC)

Ross Fenning¹ Huseyin Dogan² Keith Phalp²

BBC, UK

Ross.Fenning@bbc.co.uk

Bournemouth University, UK

{hdogan,kphalp}@bournemouth.ac.uk

2014-06-17

About Me

About Me

- Senior Software Engineer

About Me

- Senior Software Engineer
- BBC Future Media

About Me

- Senior Software Engineer
- BBC Future Media
- Content Discovery: Search

About Me

- Senior Software Engineer
- BBC Future Media
- Content Discovery: Search
- BBC Academy and University of Bradford, UK

About Me

- Senior Software Engineer
- BBC Future Media
- Content Discovery: Search
- BBC Academy and University of Bradford, UK
- Bournemouth University, UK and Lancaster University

About Me

- Senior Software Engineer
- BBC Future Media
- Content Discovery: Search
- BBC Academy and University of Bradford, UK
- Bournemouth University, UK and Lancaster University
- School of Design Engineering & Computing at Bournemouth University

What is BBC Search?

i.e. What do I work on?

The screenshot shows the BBC website homepage. At the top, the BBC logo is on the left, and navigation links for Sign in, News, Sport, Weather, iPlayer, TV, Radio, and More are in the center. A search bar is located to the right of the 'More' link, circled in red. Below the navigation bar, the main content area features a large weather forecast for London on Friday, 12 April, with a temperature of 6°C. To the right of the weather, there are three main news stories: 'Pillow fight! Striking shots from the world this week', 'US in new N. Korea missile warning', and 'The Masters'. Below these stories, there are sections for 'BBC NEWS', 'LOCAL NEWS', 'BBC SPORT', and 'FOOTBALL'. At the bottom of the page, there is a navigation bar with links for BBC now, Entertainment, News, Lifestyle, Knowledge, and Sport. Below this bar, there are three large buttons: 'Most popular', 'What's on', and 'Explore', each with a right-pointing arrow.

BBC Sign in News Sport Weather iPlayer TV Radio More **Search**

LONDON
FRIDAY, 12 APRIL
Fri 6°C Min. night Sat 6°C Sun 16°C 10°C
Change your location | 5 day forecast

Pillow fight! Striking shots from the world this week

US in new N. Korea missile warning

The Masters

BBC NEWS

- Portugal and Ireland loans extended
- BBC defends Thatcher song row move
- Thatcher 'death tweet' officer quits
- Terror suspect restrictions relaxed

LOCAL NEWS | LONDON

- Fourth arrest over gun murder
- Man jailed for slashing two women
- Baby P man sent back to prison

BBC SPORT

- Barcelona & Real Madrid kept apart
- McDermott appointed Leeds manager
- Sale Sharks 32-9 Gloucester
- Lewis Hamilton: I've deserved every...

FOOTBALL

- Benitez: Chelsea to learn from past
- Leicester 2-2 Birmingham
- Accrington 2-0 Morecambe

BBC TWO | FACTUAL
Isaac Newton: The Last Ma...

BBC RADIO 4 | COMEDY
The News Quiz
Episode 1

BBC now Entertainment News Lifestyle Knowledge Sport

Most popular > What's on > Explore >

Why BBC Search?

Isn't Google good enough?

Why BBC Search?

Isn't Google good enough?

- We can search the BBC via Google

Why BBC Search?

Isn't Google good enough?

- We can search the BBC via Google
- What can the BBC offer that Google cannot?

Why BBC Search?

Isn't Google good enough?

- We can search the BBC via Google
- What can the BBC offer that Google cannot?
- Domain knowledge about our own content

Why BBC Search?

Isn't Google good enough?

- We can search the BBC via Google
- What can the BBC offer that Google cannot?
- Domain knowledge about our own content
- More “real time” updates of content

Design Goals

Motivation

Design Goals

Motivation

- Start high-level thinking and design of a new search application

Design Goals

Motivation

- Start high-level thinking and design of a new search application
- Take a “fresh” approach as if we are replacing the whole system

Design Goals

Motivation

- Start high-level thinking and design of a new search application
- Take a “fresh” approach as if we are replacing the whole system
- A new design can (hopefully) inform iterative improvements on a current system

Design Goals

Motivation

- Start high-level thinking and design of a new search application
- Take a “fresh” approach as if we are replacing the whole system
- A new design can (hopefully) inform iterative improvements on a current system
- Existing search application serves a wide variety of use cases

Design Goals

Motivation

- Start high-level thinking and design of a new search application
- Take a “fresh” approach as if we are replacing the whole system
- A new design can (hopefully) inform iterative improvements on a current system
- Existing search application serves a wide variety of use cases
- The Search application should integrate where possible with other existing systems

Design Goals

Motivation

- Start high-level thinking and design of a new search application
- Take a “fresh” approach as if we are replacing the whole system
- A new design can (hopefully) inform iterative improvements on a current system
- Existing search application serves a wide variety of use cases
- The Search application should integrate where possible with other existing systems
- Can SSM give us contextualisation so any modelling and design reflect the real world more appropriately?

Design Goals

Motivation

- Start high-level thinking and design of a new search application
- Take a “fresh” approach as if we are replacing the whole system
- A new design can (hopefully) inform iterative improvements on a current system
- Existing search application serves a wide variety of use cases
- The Search application should integrate where possible with other existing systems
- Can SSM give us contextualisation so any modelling and design reflect the real world more appropriately?
- Can SSM helps us consider the holistic search system that emerges after this integration?

Design Goals

Motivation

- Start high-level thinking and design of a new search application
- Take a “fresh” approach as if we are replacing the whole system
- A new design can (hopefully) inform iterative improvements on a current system
- Existing search application serves a wide variety of use cases
- The Search application should integrate where possible with other existing systems
- Can SSM give us contextualisation so any modelling and design reflect the real world more appropriately?
- Can SSM helps us consider the holistic search system that emerges after this integration?
- Can SSM ensure that the emergence properties of the whole meet the needs of the UK audience and people within the BBC?

Design Goals

Requirements

Design Goals

Requirements

- Existing search application serves a wide variety of use cases

Design Goals

Requirements

- Existing search application serves a wide variety of use cases
- The Search application should integrate where possible with other existing systems

Design Goals

Requirements

- Existing search application serves a wide variety of use cases
- The Search application should integrate where possible with other existing systems
- Website is very high traffic

Design Goals

Requirements

- Existing search application serves a wide variety of use cases
- The Search application should integrate where possible with other existing systems
- Website is very high traffic
- Different parts of the overall website are maintained by distinct teams

Design Goals

Requirements

- Existing search application serves a wide variety of use cases
- The Search application should integrate where possible with other existing systems
- Website is very high traffic
- Different parts of the overall website are maintained by distinct teams
- Varying applications, data stores and content management systems

Contextualisation of the problems space through SSM

Problem Contextualisation through SSM

Problem Contextualisation through SSM

- Claim: “Hard systems” (e.g. UML) take an ontological view:

Problem Contextualisation through SSM

- Claim: “Hard systems” (e.g. UML) take an ontological view:
 - What components are there?

Problem Contextualisation through SSM

- Claim: “Hard systems” (e.g. UML) take an ontological view:
 - What components are there?
 - Who are the actors?

Problem Contextualisation through SSM

- Claim: “Hard systems” (e.g. UML) take an ontological view:
 - What components are there?
 - Who are the actors?
 - What are the interactions between actors and components?

Problem Contextualisation through SSM

- Claim: “Hard systems” (e.g. UML) take an ontological view:
 - What components are there?
 - Who are the actors?
 - What are the interactions between actors and components?
 - What are the computations or workflows steps/stages?

Problem Contextualisation through SSM

- Claim: “Hard systems” (e.g. UML) take an ontological view:
 - What components are there?
 - Who are the actors?
 - What are the interactions between actors and components?
 - What are the computations or workflows steps/stages?
- Claim: SSM takes an epistemological view:

Problem Contextualisation through SSM

- Claim: “Hard systems” (e.g. UML) take an ontological view:
 - What components are there?
 - Who are the actors?
 - What are the interactions between actors and components?
 - What are the computations or workflows steps/stages?
- Claim: SSM takes an epistemological view:
 - How does the holistic search system transform user needs to content and information?

Problem Contextualisation through SSM

- Claim: “Hard systems” (e.g. UML) take an ontological view:
 - What components are there?
 - Who are the actors?
 - What are the interactions between actors and components?
 - What are the computations or workflows steps/stages?
- Claim: SSM takes an epistemological view:
 - How does the holistic search system transform user needs to content and information?
 - How can the purposeful activity be monitored?

Problem Contextualisation through SSM

- Claim: “Hard systems” (e.g. UML) take an ontological view:
 - What components are there?
 - Who are the actors?
 - What are the interactions between actors and components?
 - What are the computations or workflows steps/stages?
- Claim: SSM takes an epistemological view:
 - How does the holistic search system transform user needs to content and information?
 - How can the purposeful activity be monitored?
 - How is the activity observed by different observers?

Problem Contextualisation through SSM

- Claim: “Hard systems” (e.g. UML) take an ontological view:
 - What components are there?
 - Who are the actors?
 - What are the interactions between actors and components?
 - What are the computations or workflows steps/stages?
- Claim: SSM takes an epistemological view:
 - How does the holistic search system transform user needs to content and information?
 - How can the purposeful activity be monitored?
 - How is the activity observed by different observers?
 - How can different observers inform the activity?

Problem Contextualisation through SSM

- Claim: “Hard systems” (e.g. UML) take an ontological view:
 - What components are there?
 - Who are the actors?
 - What are the interactions between actors and components?
 - What are the computations or workflows steps/stages?
- Claim: SSM takes an epistemological view:
 - How does the holistic search system transform user needs to content and information?
 - How can the purposeful activity be monitored?
 - How is the activity observed by different observers?
 - How can different observers inform the activity?
- Interactive Management (Dogan and Henshaw, 2010) can capture requirements and transition to formal models

Problem Contextualisation through SSM

Rich Picture

Use Case Extraction

Identifying users

Use Case Extraction

Identifying users

- Identify all users of the system in the rich picture

Use Case Extraction

Identifying users

- Identify all users of the system in the rich picture
- Separate *direct* users from *meta-level* users

Use Case Extraction

Identifying users

- Identify all users of the system in the rich picture
- Separate *direct* users from *meta-level* users
- Direct use cases should emerge from the rich picture

Use Case Extraction

Identifying users

- Identify all users of the system in the rich picture
- Separate *direct* users from *meta-level* users
- Direct use cases should emerge from the rich picture
- NOTE: This separation is still *subjective* and use cases may change as the rich picture changes

Use Case Extraction

Meta-level Users

Use Case Extraction

Meta-level Users

- Meta-level users do not transition to the use case models directly

Use Case Extraction

Meta-level Users

- Meta-level users do not transition to the use case models directly
- *Systems Thinking* tells us to consider how meta-level users are affected by emergent properties of the holistic system

Use Case Extraction

Direct Users

Use Case Extraction

Direct Users

- Direct users transition directly to actors into a UML use case diagram

Use Case Extraction

Direct Users

- Direct users transition directly to actors into a UML use case diagram
- Use cases should derive from and be justifiable by the rich picture

Use Case Extraction

Direct Users

- Direct users transition directly to actors into a UML use case diagram
- Use cases should derive from and be justifiable by the rich picture
- Direct users divide up into *content production* and *audience*

Use Case Extraction

Direct Users

- Direct users transition directly to actors into a UML use case diagram
- Use cases should derive from and be justifiable by the rich picture
- Direct users divide up into *content production* and *audience*
- Examples of each:

Use Case Extraction

Direct Users

- Direct users transition directly to actors into a UML use case diagram
- Use cases should derive from and be justifiable by the rich picture
- Direct users divide up into *content production* and *audience*
- Examples of each:
 - Producers of TV programmes metadata

Use Case Extraction

Direct Users

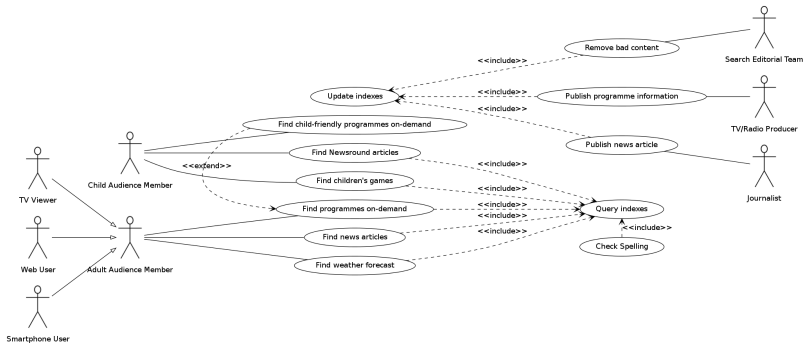
- Direct users transition directly to actors into a UML use case diagram
- Use cases should derive from and be justifiable by the rich picture
- Direct users divide up into *content production* and *audience*
- Examples of each:
 - Producers of TV programmes metadata
 - Users of the search web application

Use Case Extraction

UML Use Case Diagram

Use Case Extraction

UML Use Case Diagram

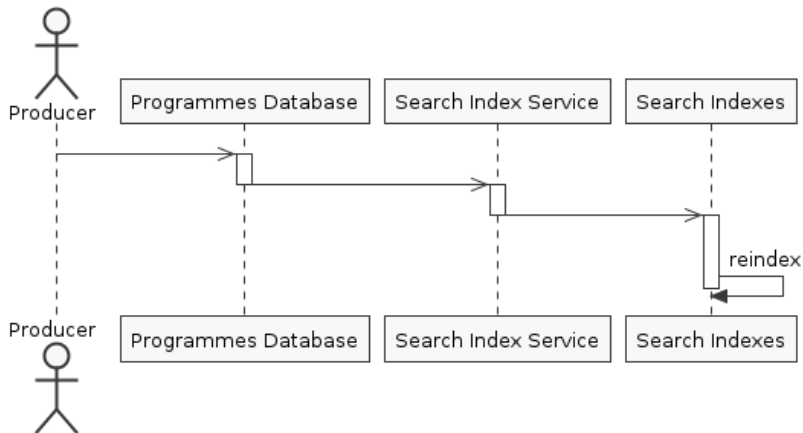


System Behaviour

Programmes metadata producer

System Behaviour

Programmes metadata producer

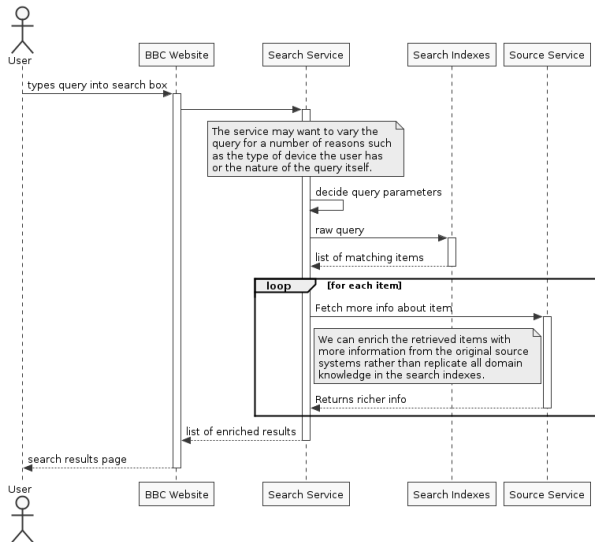


System Behaviour

Website user

System Behaviour

Website user

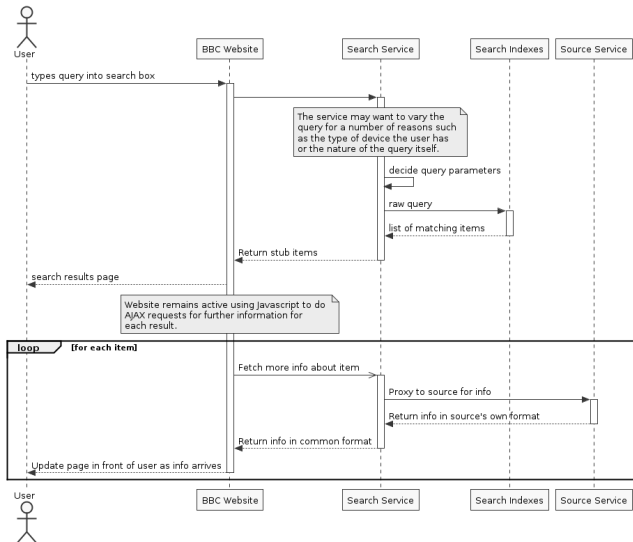


System Behaviour

Lazy-loading page via AJAX

System Behaviour

Lazy-loading page via AJAX



Domain Modelling

Domain Modelling

- Use cases identify types of content users want to search

Domain Modelling

- Use cases identify types of content users want to search
- Ontology designs exist for specific content domains, e.g. programmes, sport

Domain Modelling

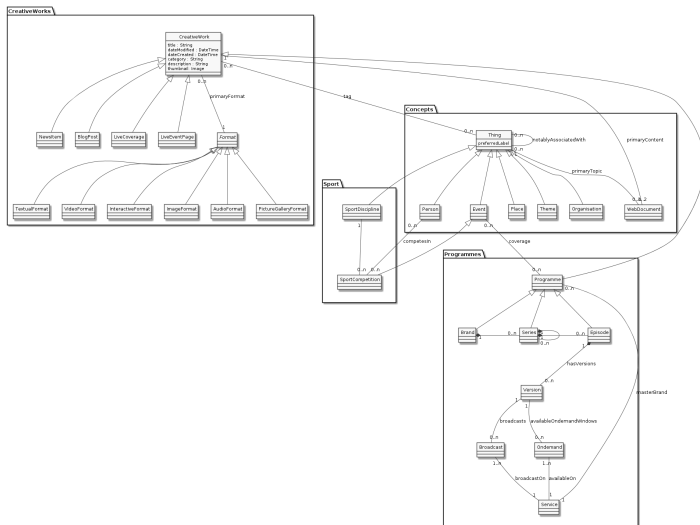
- Use cases identify types of content users want to search
- Ontology designs exist for specific content domains, e.g. programmes, sport
- A search across all content types needs a model to span all domains

Domain Modelling

Class Diagram

Domain Modelling

Class Diagram



Discussion

Use Cases

Discussion

Use Cases

- Use case model seems justifiable at a high level

Discussion

Use Cases

- Use case model seems justifiable at a high level
- Many use cases and variations are missing

Discussion

Use Cases

- Use case model seems justifiable at a high level
- Many use cases and variations are missing
- More collaborative and continuous use of SSM should incorporate respective domain experts

Discussion

Use Cases

- Use case model seems justifiable at a high level
- Many use cases and variations are missing
- More collaborative and continuous use of SSM should incorporate respective domain experts
- More user testing, audience feedback, requirements-gathering, etc. needed

Discussion

Use Cases

- Use case model seems justifiable at a high level
- Many use cases and variations are missing
- More collaborative and continuous use of SSM should incorporate respective domain experts
- More user testing, audience feedback, requirements-gathering, etc. needed
- Possible need to iterate SSM processes towards a collection of use case models

Discussion

System Behaviour

Discussion

System Behaviour

- Enterprise Integration patterns

Discussion

System Behaviour

- Enterprise Integration patterns
 - Asynchronous messaging

Discussion

System Behaviour

- Enterprise Integration patterns
 - Asynchronous messaging
 - Content Enricher – Replicate information from other systems redundantly

Discussion

System Behaviour

- Enterprise Integration patterns
 - Asynchronous messaging
 - Content Enricher – Replicate information from other systems redundantly
 - Claim Check – Store absolute minimum information to function and maintain references to source systems

Discussion

System Behaviour

- Enterprise Integration patterns
 - Asynchronous messaging
 - Content Enricher – Replicate information from other systems redundantly
 - Claim Check – Store absolute minimum information to function and maintain references to source systems
 - Index vs. warehouse?

Discussion

System Behaviour

- Enterprise Integration patterns
 - Asynchronous messaging
 - Content Enricher – Replicate information from other systems redundantly
 - Claim Check – Store absolute minimum information to function and maintain references to source systems
 - Index vs. warehouse?
 - How to evaluate the better approach?

Discussion

Domain Model

Discussion

Domain Model

- Large, fully-inclusive data model useful mainly as communication artefact

Discussion

Domain Model

- Large, fully-inclusive data model useful mainly as communication artefact
- Aggregate models hard to keep in sync with changes in respective domains

Discussion

Domain Model

- Large, fully-inclusive data model useful mainly as communication artefact
- Aggregate models hard to keep in sync with changes in respective domains
- We can extract a polymorphic supertype containing only properties needed for a search application

Discussion

Domain Model

- Large, fully-inclusive data model useful mainly as communication artefact
- Aggregate models hard to keep in sync with changes in respective domains
- We can extract a polymorphic supertype containing only properties needed for a search application
- We can go further and model a “Search Result” facade

Discussion

Schemalessness and Duck Typing

Discussion

Schemalessness and Duck Typing

- Statically typed approach: A *Place* always has a “today’s weather forecast” attribute

Discussion

Schemalessness and Duck Typing

- Statically typed approach: A *Place* always has a “today’s weather forecast” attribute
- Dynamically (“duck”) typed approach: any entity can have a “today’s weather forecast” attribute and we display this forecast if that attribute is present

Discussion

Schemalessness and Duck Typing

- Statically typed approach: A *Place* always has a “today’s weather forecast” attribute
- Dynamically (“duck”) typed approach: any entity can have a “today’s weather forecast” attribute and we display this forecast if that attribute is present
- Allows other entities in future to have forecasts too (e.g. *Events* such as football matches)

Discussion

Schemalessness and Duck Typing

- Statically typed approach: A *Place* always has a “today’s weather forecast” attribute
- Dynamically (“duck”) typed approach: any entity can have a “today’s weather forecast” attribute and we display this forecast if that attribute is present
- Allows other entities in future to have forecasts too (e.g. *Events* such as football matches)
- Search application can store all fields from upstream sources and choose when to start using them

Discussion

Overall Model

Discussion

Overall Model

- Promising at a high level

Discussion

Overall Model

- Promising at a high level
- Agile methodology would encourage developing prototype or MVP to test assumptions early

Discussion

Overall Model

- Promising at a high level
- Agile methodology would encourage developing prototype or MVP to test assumptions early
- Lacking suitable “monitoring” of the system as taught by SSM, e.g. Checkland’s “3 Es” (efficacy, efficiency, effectiveness)

Evaluation and Difficulties

Non-functional Requirements

Evaluation and Difficulties

Non-functional Requirements

- BBC is a high-traffic website

Evaluation and Difficulties

Non-functional Requirements

- BBC is a high-traffic website
- Content such as news articles are published frequently

Evaluation and Difficulties

Non-functional Requirements

- BBC is a high-traffic website
- Content such as news articles are published frequently
- How to model requirements around performance, robustness and latency?

Evaluation and Difficulties

Detailed Design and Agile

Evaluation and Difficulties

Detailed Design and Agile

- Models developed at such a high level lack detailed design (not surprisingly)

Evaluation and Difficulties

Detailed Design and Agile

- Models developed at such a high level lack detailed design (not surprisingly)
- Agile methodologies encourage designing and delivering a small part of the system early

Evaluation and Difficulties

Detailed Design and Agile

- Models developed at such a high level lack detailed design (not surprisingly)
- Agile methodologies encourage designing and delivering a small part of the system early
- Detailed design emerges in response to feedback and change

Evaluation and Difficulties

Detailed Design and Agile

- Models developed at such a high level lack detailed design (not surprisingly)
- Agile methodologies encourage designing and delivering a small part of the system early
- Detailed design emerges in response to feedback and change
- Hard models can be altered in time with these iterations

Evaluation and Difficulties

Detailed Design and Agile

- Models developed at such a high level lack detailed design (not surprisingly)
- Agile methodologies encourage designing and delivering a small part of the system early
- Detailed design emerges in response to feedback and change
- Hard models can be altered in time with these iterations
- We can defer outstanding questions about hard models until we might have more information

Evaluation and Difficulties

Detailed Design and Agile

- Models developed at such a high level lack detailed design (not surprisingly)
- Agile methodologies encourage designing and delivering a small part of the system early
- Detailed design emerges in response to feedback and change
- Hard models can be altered in time with these iterations
- We can defer outstanding questions about hard models until we might have more information
- This seems in line with the continuous feedback cycles within SSM

Further Work

Further Work

- How can the models be effectively validated?

Further Work

- How can the models be effectively validated?
- Can the rich picture and use of SSM be repeated and expanded with input from a wider range of experts?

Further Work

- How can the models be effectively validated?
- Can the rich picture and use of SSM be repeated and expanded with input from a wider range of experts?
- How will the models change when this happens?

Further Work

- How can the models be effectively validated?
- Can the rich picture and use of SSM be repeated and expanded with input from a wider range of experts?
- How will the models change when this happens?
- Are there more parallels between Agile and SSM that engineers in industry have yet to appreciate?

Further Work

- How can the models be effectively validated?
- Can the rich picture and use of SSM be repeated and expanded with input from a wider range of experts?
- How will the models change when this happens?
- Are there more parallels between Agile and SSM that engineers in industry have yet to appreciate?
- Should more development processes be considered alongside SSM and UML? e.g. RUP (OpenUP) or AUP?