



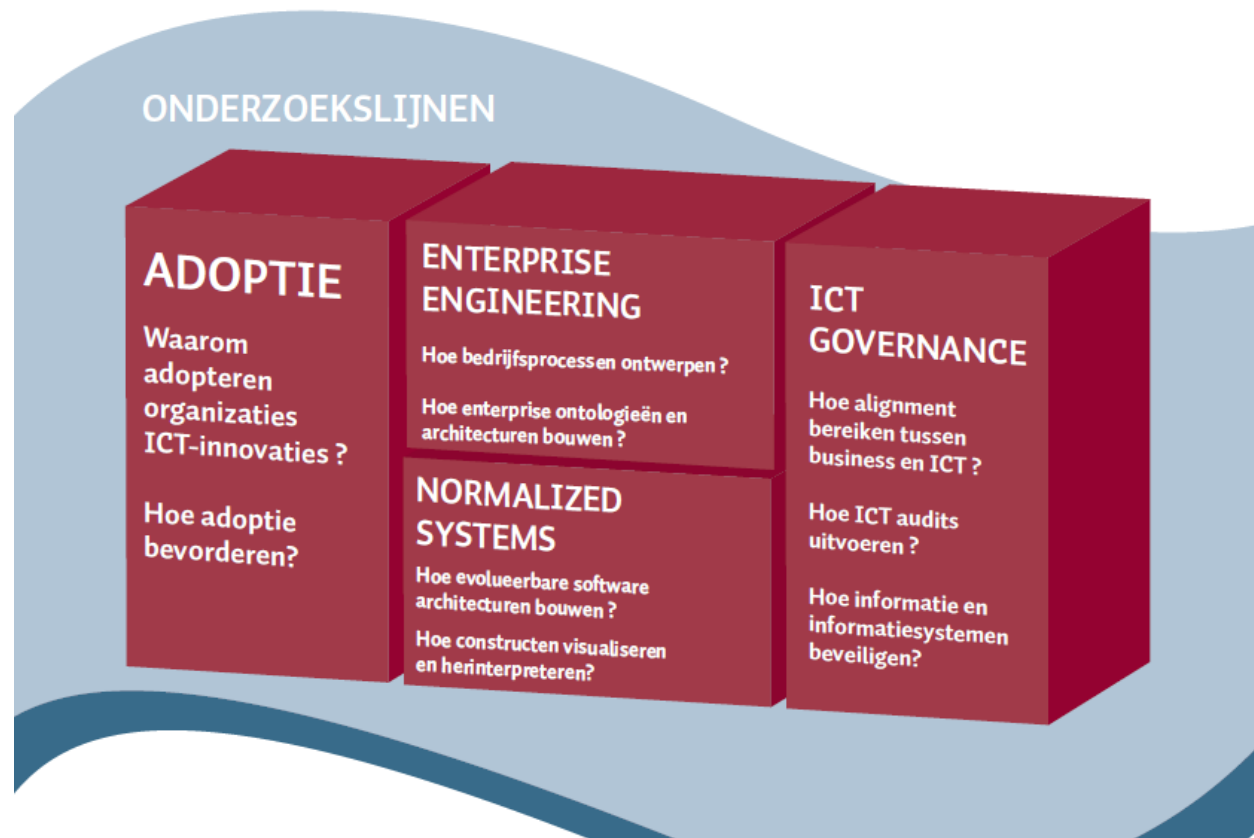
# Normalized Systems: Van academische theorie tot open innovatie alliantie

Prof. dr. Herwig Mannaert

Prof. dr. Jan Verelst



# Departement Beleidsinformatica





# Overview

- Inleiding
- Normalized Systems Theorie
  - Fundamenten: stabiliteit en entropie
  - Normalized Systems en software
  - Normalized Systems en bedrijfsprocessen
    - Enterprise Engineering
- Alliantie voor Open Innovatie
  - Technologie en innovatie
  - Normalized Systems Alliance



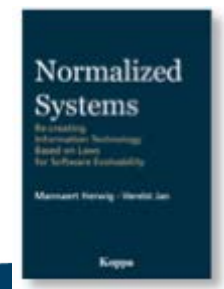
# Overview

- Inleiding
- Normalized Systems Theorie
  - Fundamenten: stabiliteit en entropie
  - Normalized Systems en software
  - Normalized Systems en bedrijfsprocessen
    - Enterprise Engineering
- Alliantie voor Open Innovatie
  - Technologie en innovatie
  - Normalized Systems Alliance



# About NS Theory

- A theoretical framework investigating **Modular Structures** under **Change**
  - Based on Systems Theoretic concepts
    - → Completely independent of any framework, programming language, package, ...
    - Has shown to be able to deal with the challenge of increasing complexity
      - E.g. hardware, Internet, space industry...
  - Initial scope: Modular Structures in Software Architectures
  - Publications: book, >50 conference proceedings, (invited) lectures at different universities...
  - Education: undergraduate, postgraduate...





## *The Dream:* Doug Mc Ilroy



"expect families of routines to be constructed on *rational principles* so that families fit together as **building blocks**. In short, [the user] should be able safely to regard components as black boxes."

uit: McIlroy, *Mass Produced Software Components*,  
1968 NATO Conference on Software Engineering, Garmisch, Germany.



# *The Reality: Manny Lehman*

## **The Law of Increasing Complexity Manny Lehman**

*“As an evolving program is continually changed, its complexity, reflecting deteriorating structure, increases unless work is done to maintain or reduce it.”*

*Proceedings of the IEEE, vol. 68, nr. 9, september 1980, pp. 1068.*



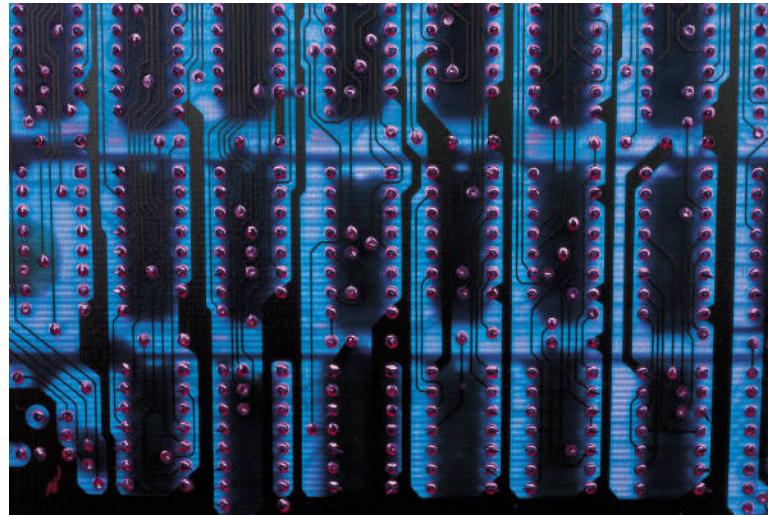
# Overview

- Inleiding
- Normalized Systems Theorie
  - Fundamenten: stabiliteit en entropie
  - Normalized Systems en software
  - Normalized Systems en bedrijfsprocessen
    - Enterprise Engineering
- Alliantie voor Open Innovatie
  - Technologie en innovatie
  - Normalized Systems Alliance





# Controlling Complexity can be done...



Other disciplines have mastered the **structured assembly of large numbers of fine-grained static** modules... e.g. hardware !

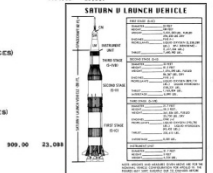
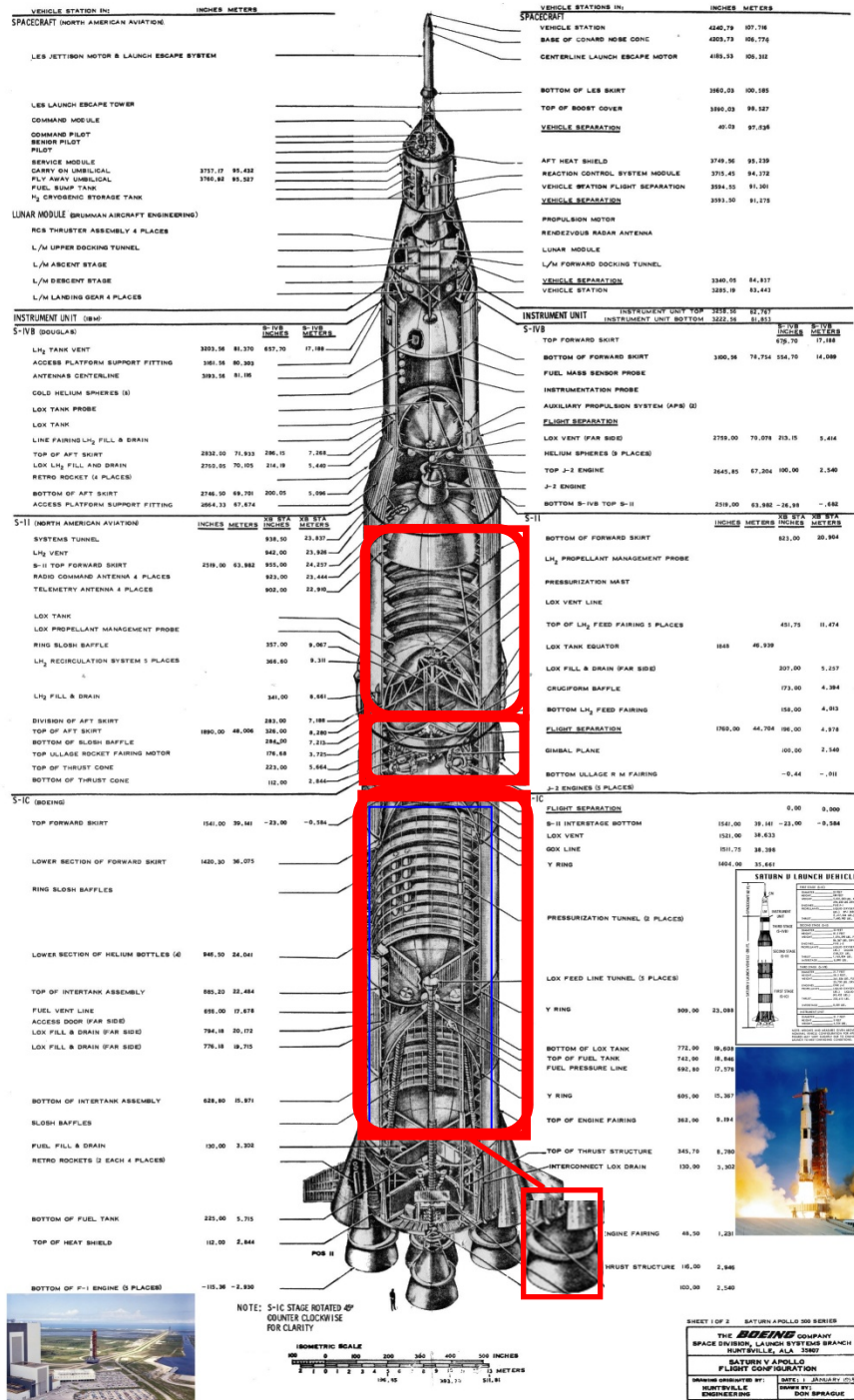




# However:

# Modularity is static

## Universiteit Antwerpen



SHEET 1 OF 2 SATURN V BURNER SERIES  
 THE ASTERION COMPANY  
 SPACE DESIGN, LAUNCH SYSTEMS BRANCH  
 HUNTSVILLE, ALA. 35897  
 SATURN V ROLL-OUT  
 FLIGHT CONFIGURATION  
 DRAWN BY:  
 ENGINEERING  
 CHECKED BY:  
 DESIGN





## Some more examples ...

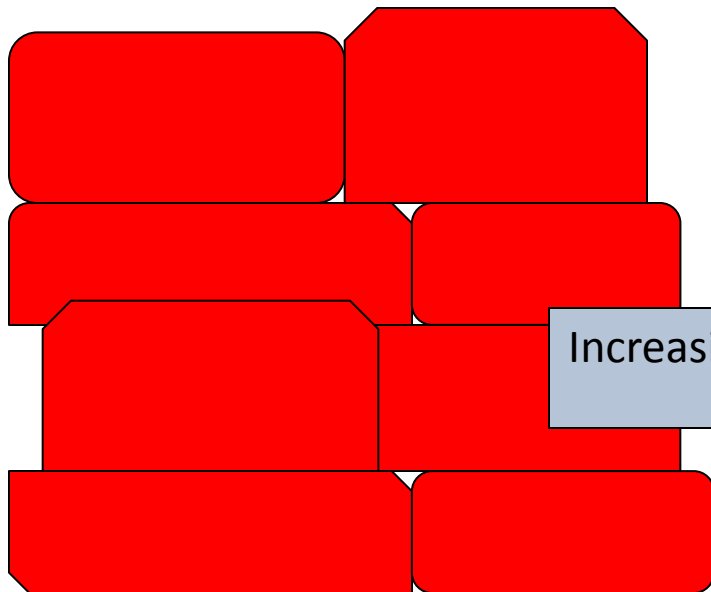
- Airbus 380 could not be designed by taking x2 for every measure of the Airbus 340 plan
- Instruction set of a microprocessor cannot be extended by adding another module
- Construction buildings cannot grow over time by simply adding additional units
- Car performance cannot be upgraded by adding additional parts to the engine
- ...



# Evolvability: The main issue

Static Modularity

Complexity



Increasing Change



Lehman, No McIlroy



# Advantages of Modules

Reductie van complexiteit

Hergebruik

Evolueerbaarheid

# Systems Theory → Evolvability

- Stability in system dynamics:
  - Is used to study dynamics of system operations
    - Mechanical, e.g. constructions, vehicles, ...
    - Electrical, e.g. amplifiers, generators, ...
    - Hydraulical, e.g. pumps, engines, ...
    - ...
  - Is not used to study dynamics of system artefacts
    - Rockets and airplanes
    - Software and information systems
    - Organizations and human enterprises
    - ...



# Overview

- Inleiding
- **Normalized Systems Theorie**
  - Fundamenten: stabiliteit en entropie
  - **Normalized Systems en software**
  - Normalized Systems en bedrijfsprocessen
    - Enterprise Engineering
- Alliantie voor Open Innovatie
  - Technologie en innovatie
  - Normalized Systems Alliance



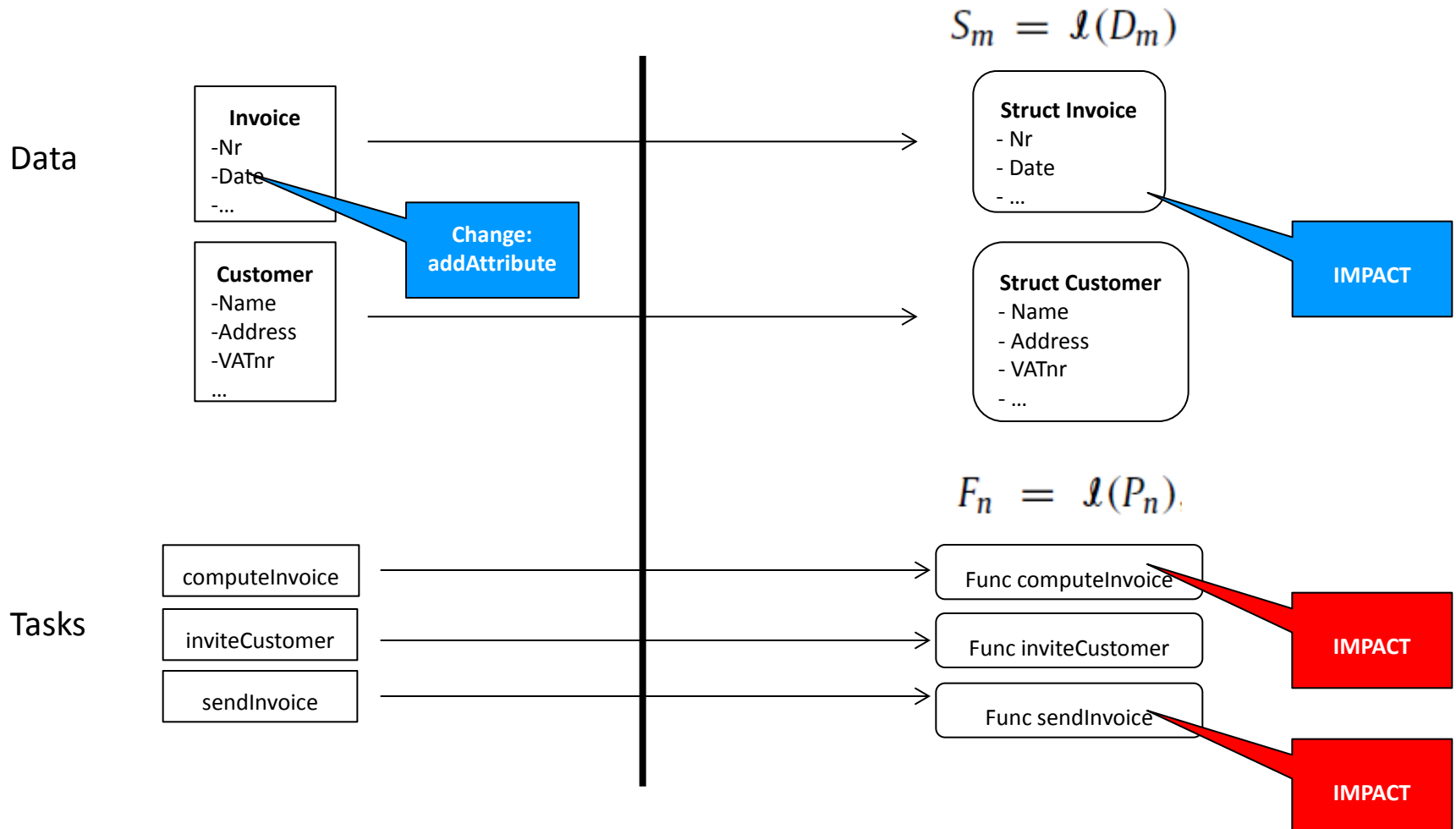


# Step 1: NS Principles

The need for fine-grained modular structures



# A Simple Transformation



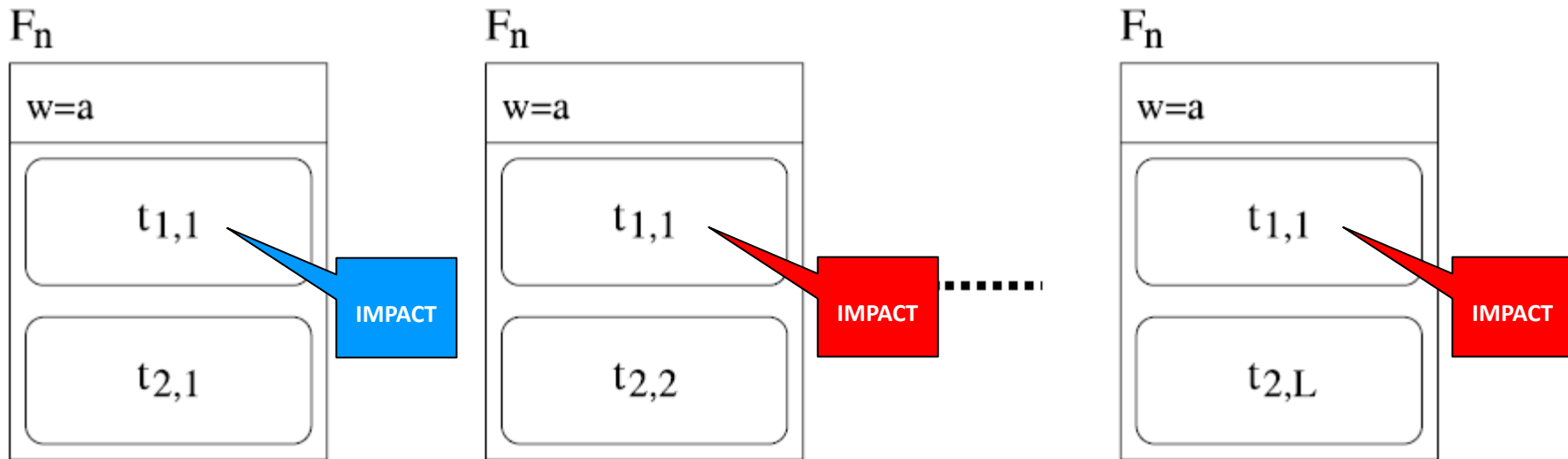


# NS Principles

- Modularity x Change → **Combinatorial Effects (CE) !**
  - CE = (hidden) **coupling** or dependencies, **increasing with size of the system !**
  - **NS Principles** identify CE at seemingly orthogonal levels
    - SoC: Which tasks do you **combine** in a single module ?
      - "An action entity can only contain a single task."
    - DVT: How do you **combine** a data and action module ?
      - "Data entities that are received as input or produced as output by action entities, need to exhibit version transparency."
    - AVT: How do you **combine** 2 modules ?
      - "Action entities that are called by other action entities, need to exhibit version transparency."
    - SoS: How do you **combine** modules in a workflow ?
      - "The calling of an action entity by another action entity needs to exhibit state keeping."



# Separations of Concerns



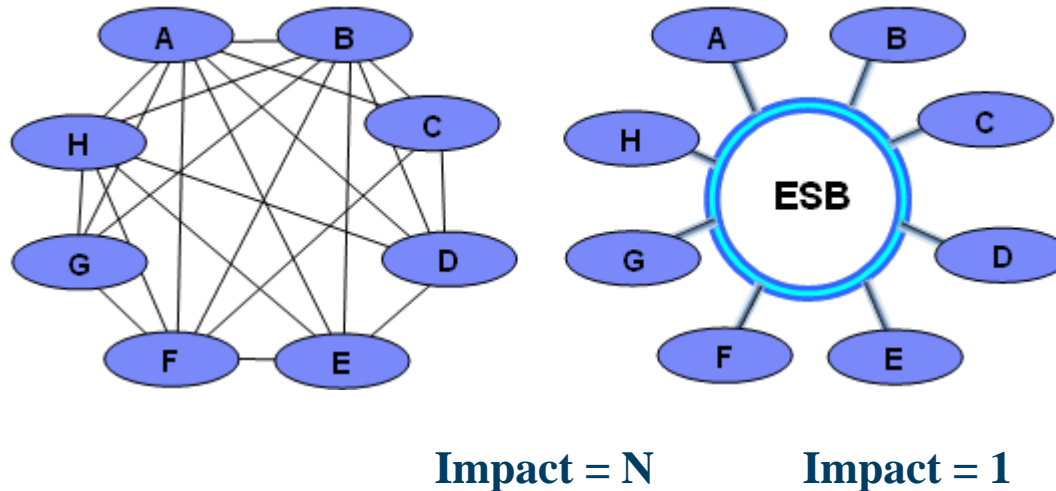
**Fig. 1.** Schematic representation of a function  $F_n$  with two tasks and multiple versions.

$$\mathcal{S}_m = \mathcal{I}(\mathcal{R}_m) \cup \{F_n(w = a, t_{1,2}, t_{2,l})\}_{l=1,\dots,L}$$



# IT: Enterprise Service Bus

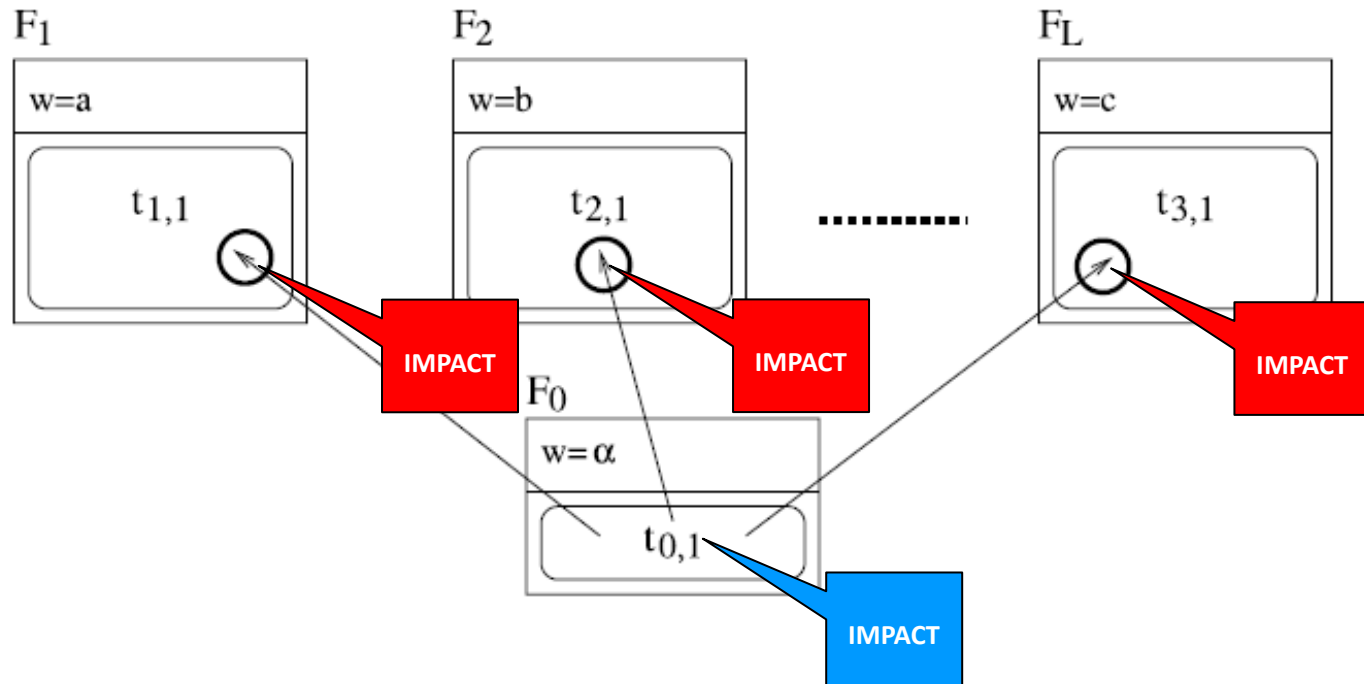
- The effort to include an additional component may or may not vary with the system size  
*or: airline spoke and hub*



Source: [http://nl.wikipedia.org/wiki/Enterprise\\_Service\\_Bus](http://nl.wikipedia.org/wiki/Enterprise_Service_Bus)



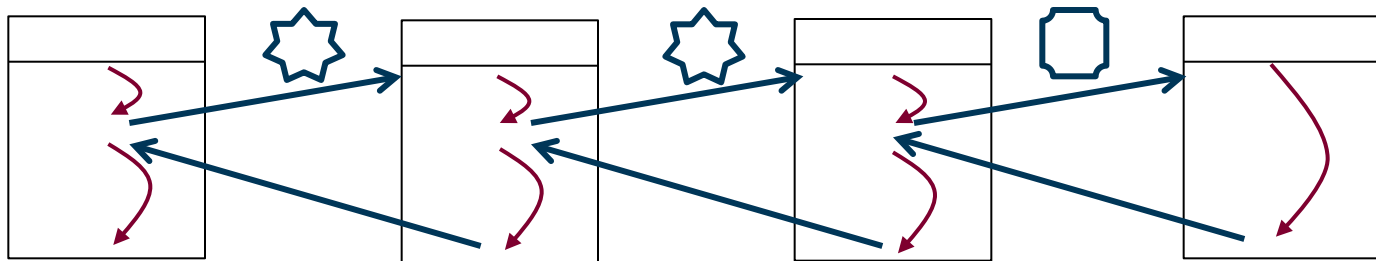
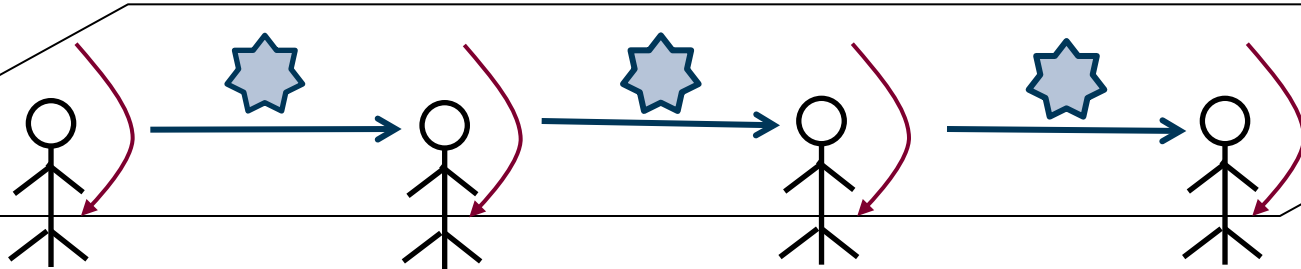
# Separation of States



$$\mathcal{S}_m = \mathcal{I}(\mathcal{R}_m) \cup \{F_l(w, t_{l,2})\}_{l=1,\dots,L}$$

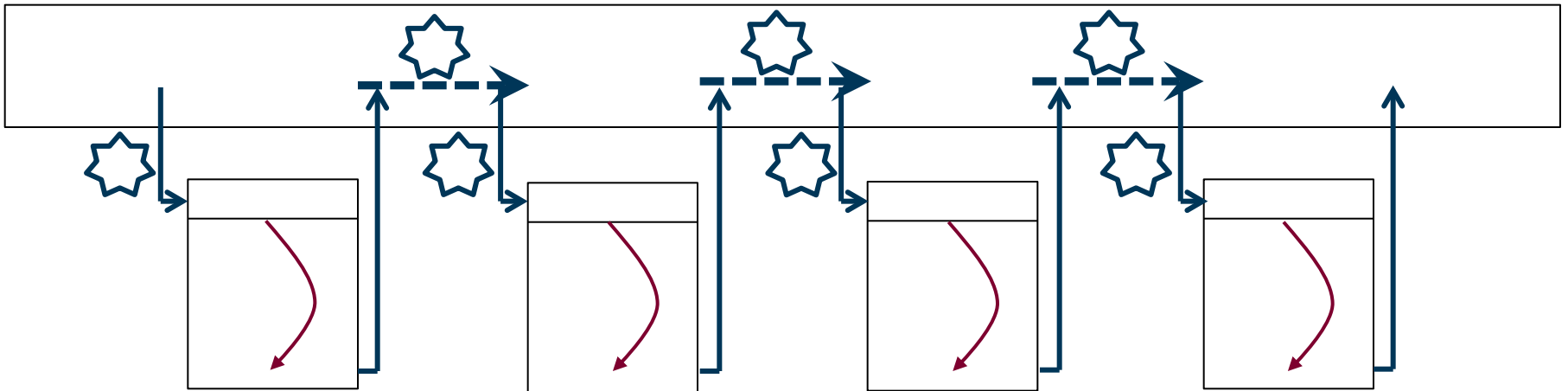
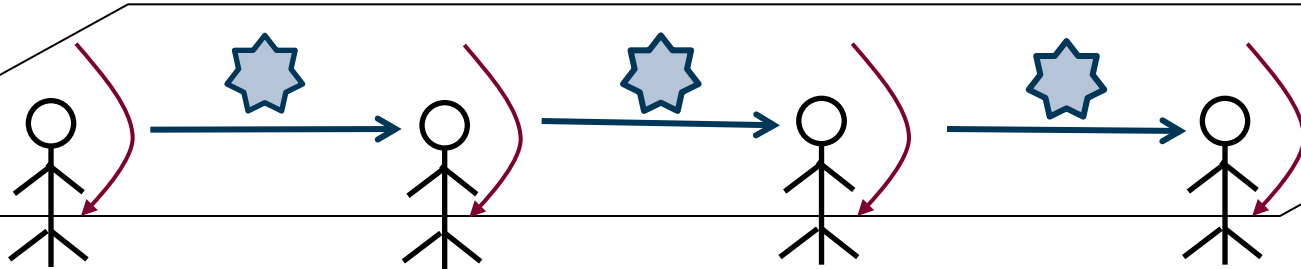


# Assembly vs. Object Pipelines





# Assembly vs. Statefull Control





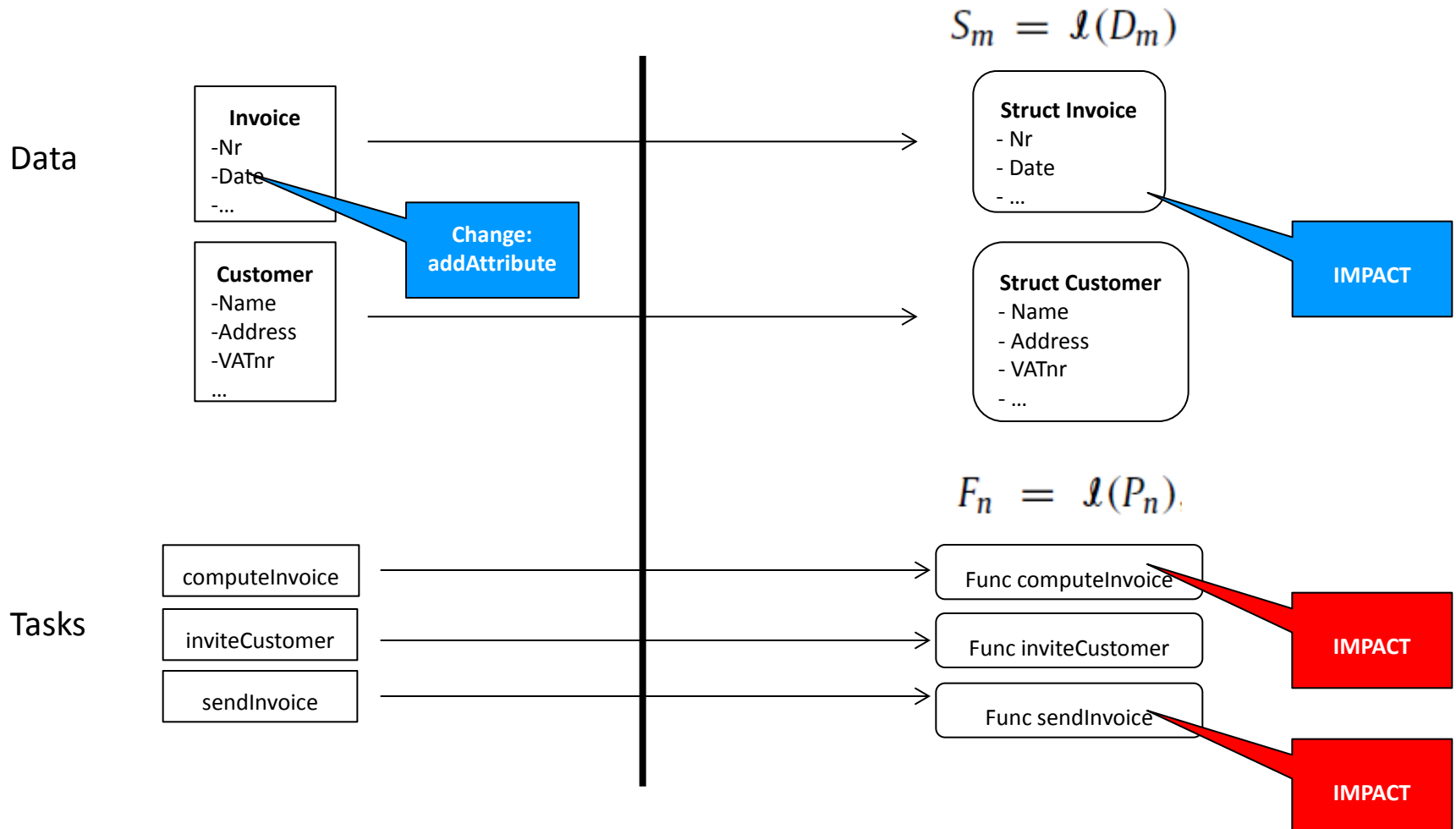


## Step 2: NS Elements

How to control large numbers of fine-grained modules, in theory and practice ?

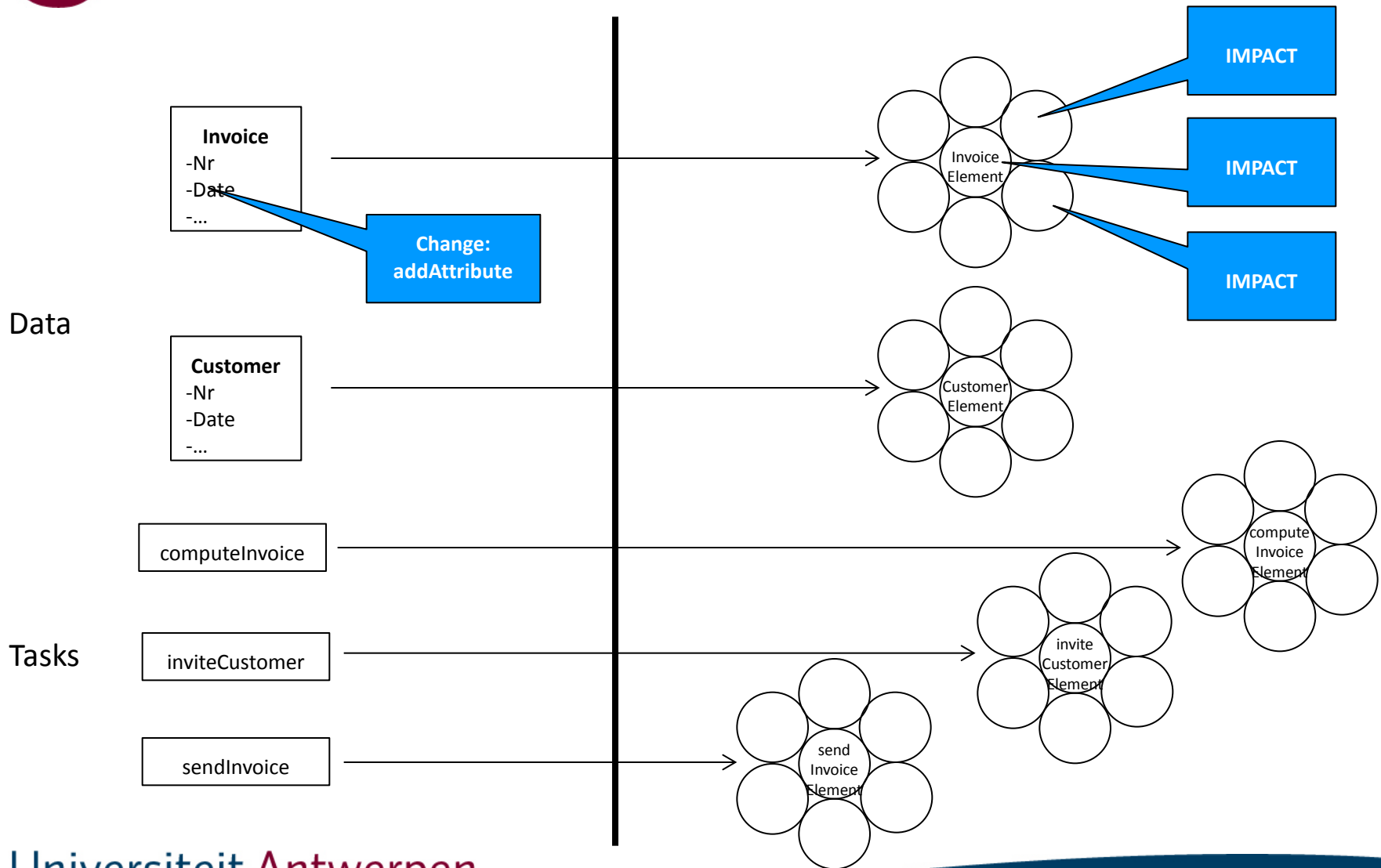


# A Simple Transformation





# A More Complex Transformation



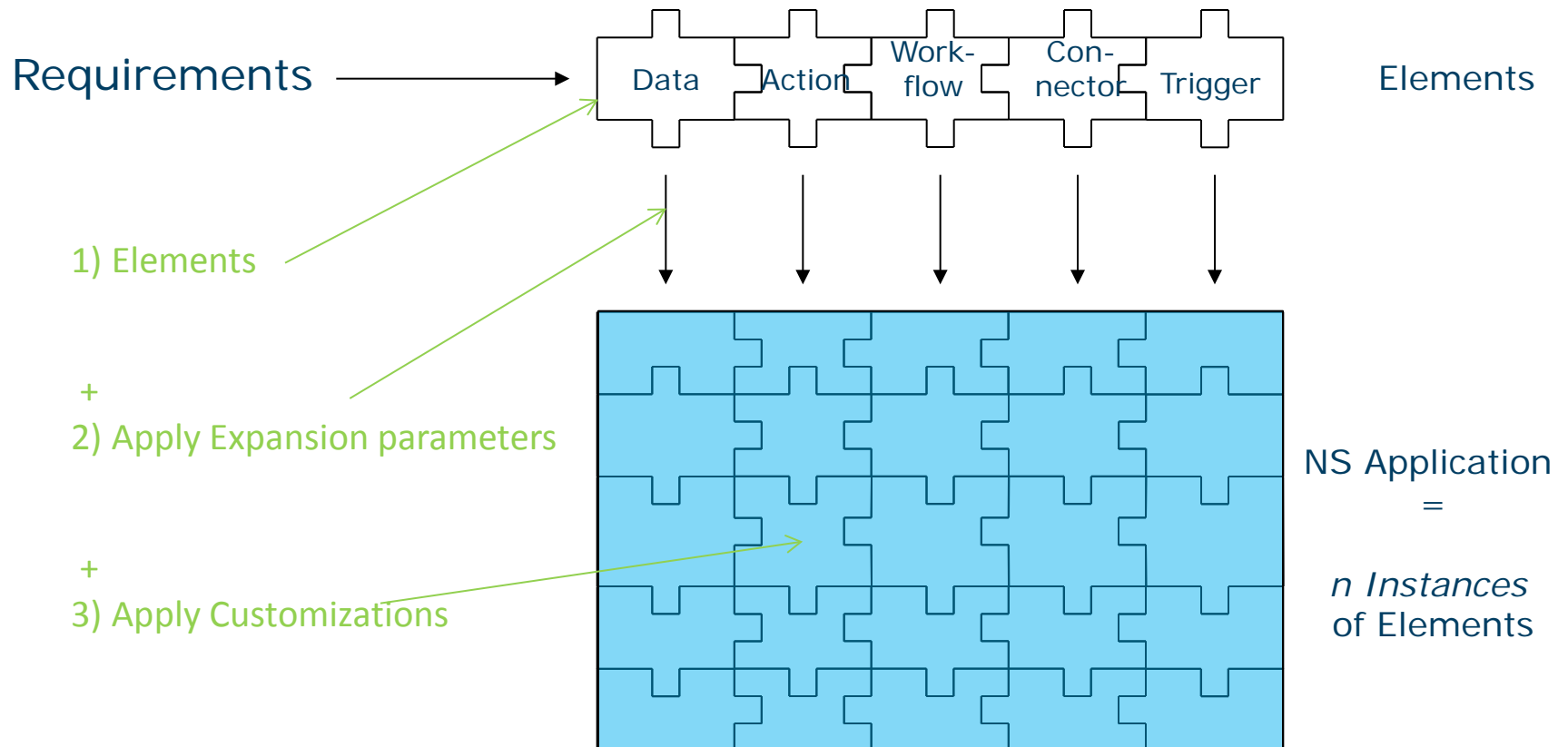


# Normalized Systems Elements

- The proposed solution =
  - Structure through Encapsulations, called Elements
    - A Java class is encapsulated in 8-10 other classes, dealing with cross-cutting concerns, in order to deal with the anticipated changes *without CE*, and fully separating the element from all other elements.
    - Every element is described by a “detailed design pattern”. Every element builds on other elements.
    - Every design pattern is executable, and can be expanded automatically.
  - Realizing the core functionality of Information Systems
- Application =  $n$  instantiations of Elements

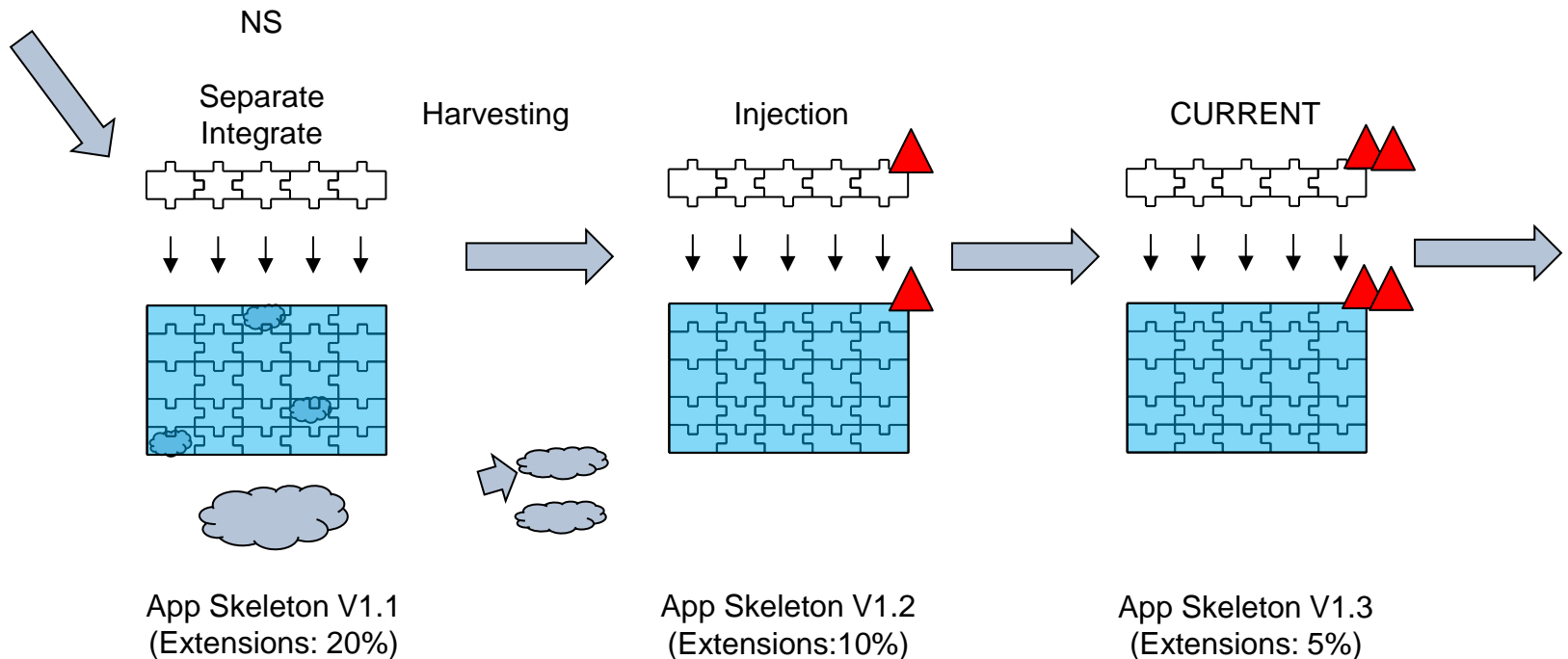
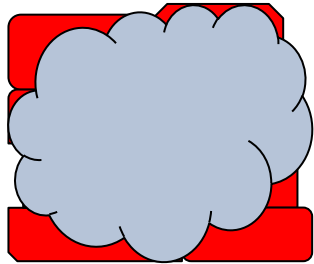


# Building using NS Elements





# Regeneration





## Google-dochter bouwt smartphone met losse 'blokjes'

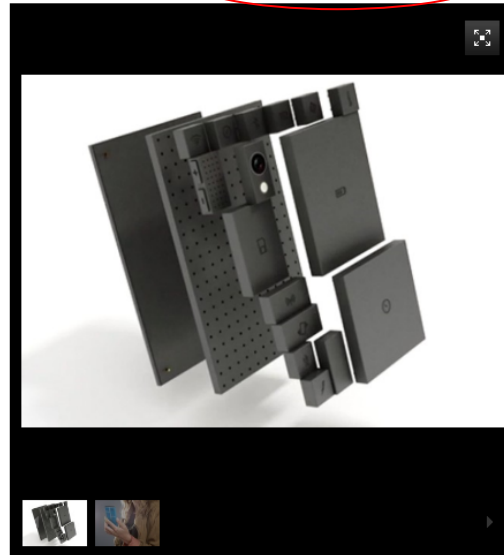


Michael Ilegems

29/10/2013 - 11:22

Commentaar (0)

Motorola schaart zich achter het concept van de 'modulaire' smartphone, een toestel dat bestaat uit losse, makkelijk te vervangen onderdelen en in principe eeuwig meegaat.



Motorola, een dochteronderneming van Google, heeft Project Ara aangekondigd; een ecosysteem voor 'modulaire' smartphones dat moet uitgroeien tot een soort "Android voor hardware".

Het idee achter modulaire smartphones is dat je als gebruiker losse telefoononderdelen of 'modules' – processor, toetsenbord, scherm, batterij – kunt aanschaffen bij verschillende fabrikanten, om hiermee een soort gepersonaliseerde smartphone samen te stellen. Als bepaalde componenten het na verloop van tijd begeven, kan je deze makkelijk vervangen door nieuwe, zodat je je toestel niet voor het minste hoeft weg te gooien. Een modulaire smartphone is in theorie eeuwig houdbaar.

Eerder bedacht de Nederlandse ontwerper Dave Hakkens met Phonebloks al een soortgelijk concept. Motorola gaat voor zijn Ara-project dan ook in zee met Hakkens. De Google-dochter wil meer bepaald een beroep doen op de community van gebruikers die de Nederlander reeds met zijn Phonebloks-concept wist te enthousiasmeren.

Motorola roept hardwareproducenten op 'modules' te ontwikkelen voor het

Voor uw financiële  
**JOB OP MAAT**



Vind uw nieuwe job via  
eFinancialCareers

**eFINANCIALCAREERS**

### Nieuwsbrief

- Schrijf u in op Data News dagelijkse nieuwsbrief
- Schrijf u in op Data News Start-ups maandelijkse nieuwsbrief
- Schrijf u in op Data News Jobs wekelijkse nieuwsbrief

E-mail:

**INSCHRIJVEN**

### Google-dochter bouwt smartphone met losse 'blokjes'

- Motorola lanceert Moto X: een Google-smartphone die je kan personaliseren
- Motorola lanceert 'heroische smartphone' in oktober
- Google schrapt 1.200 banen bij Motorola
- Google werkt aan 'onbreekbare' supersmartphone
- "Google en Motorola maken buigzame smartphone en tablet"



# Overview

- Inleiding
- Normalized Systems Theorie
  - Fundamenten: stabiliteit en entropie
  - Normalized Systems en software
  - Normalized Systems en bedrijfsprocessen
    - Enterprise Engineering
- Alliantie voor Open Innovatie
  - Technologie en innovatie
  - Normalized Systems Alliance





# Modularity at the Enterprise Level?

- Campagnolo and Camuffo (2009)
  - Point at different definitions of modularity
  - Identify 125 studies related to modularity in academic literature on Management since 1986
  - Trace modularity as a design principle for organizations back to Sanchez and Mahoney (1996)
- For example:
  - Op 't Land (2008): Organization construction rules related to modularity
  - Tewinkel et al.(2008): Modularity in Organizational structures
  - Van Ark and De Jong (2004): Modularity in Services
  - Tiwana (2008): Modularity and Organizational Knowledge
- General Indication: Organizations as modular departments, business processes consisting of modular steps, ...



# Lack of theory on BP modularity

In the Business Process Management domain, research on modularity has set off as well. Introducing modularity within business process models mostly resembles the use of sub-processes (Reijers and Mendling, 2008; Reijers et al., 2010). Sub-processes reduce the complexity of the models (Gruhn and Laue, 2009), and as a consequence enhance the model's understandability by hiding irrelevant information (Mendling et al., 2007b; Reijers et al., 2010). Identifying such modular sub-processes might be guided by selecting process modules exhibiting a single input and a single output (Basu and Blanning, 2003). Although multiple authors have already indicated the usefulness of the concept within business process design — Adler (1988), for instance, investigated the decomposition of data flow diagrams — concrete design rules to modularize business processes are still lacking (Reijers and Mendling, 2008). Therefore, modularizing business processes often happens in an ad-hoc way, indicating the need for explicit guidelines to introduce modularity within business processes (Reijers and Mendling, 2008; Reijers et al., 2010).



# Current BP contain CE

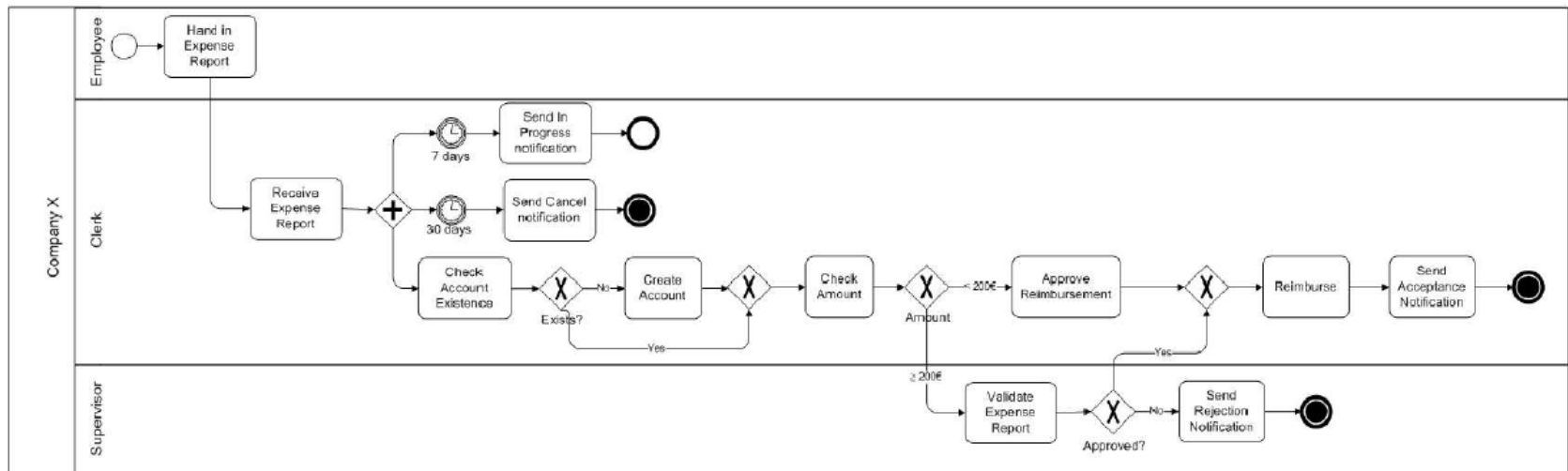


Figure 3.2: BPMN model of an expense reimbursement business process.



# Additional BP examples

- Mail virtualization

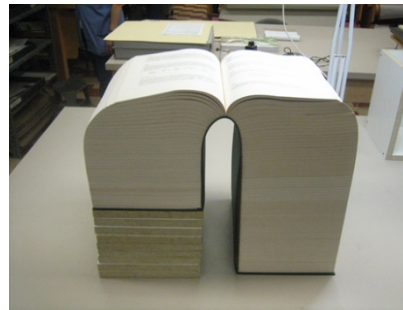


- Curriculum change

For students enrolling Fall 2009 or later

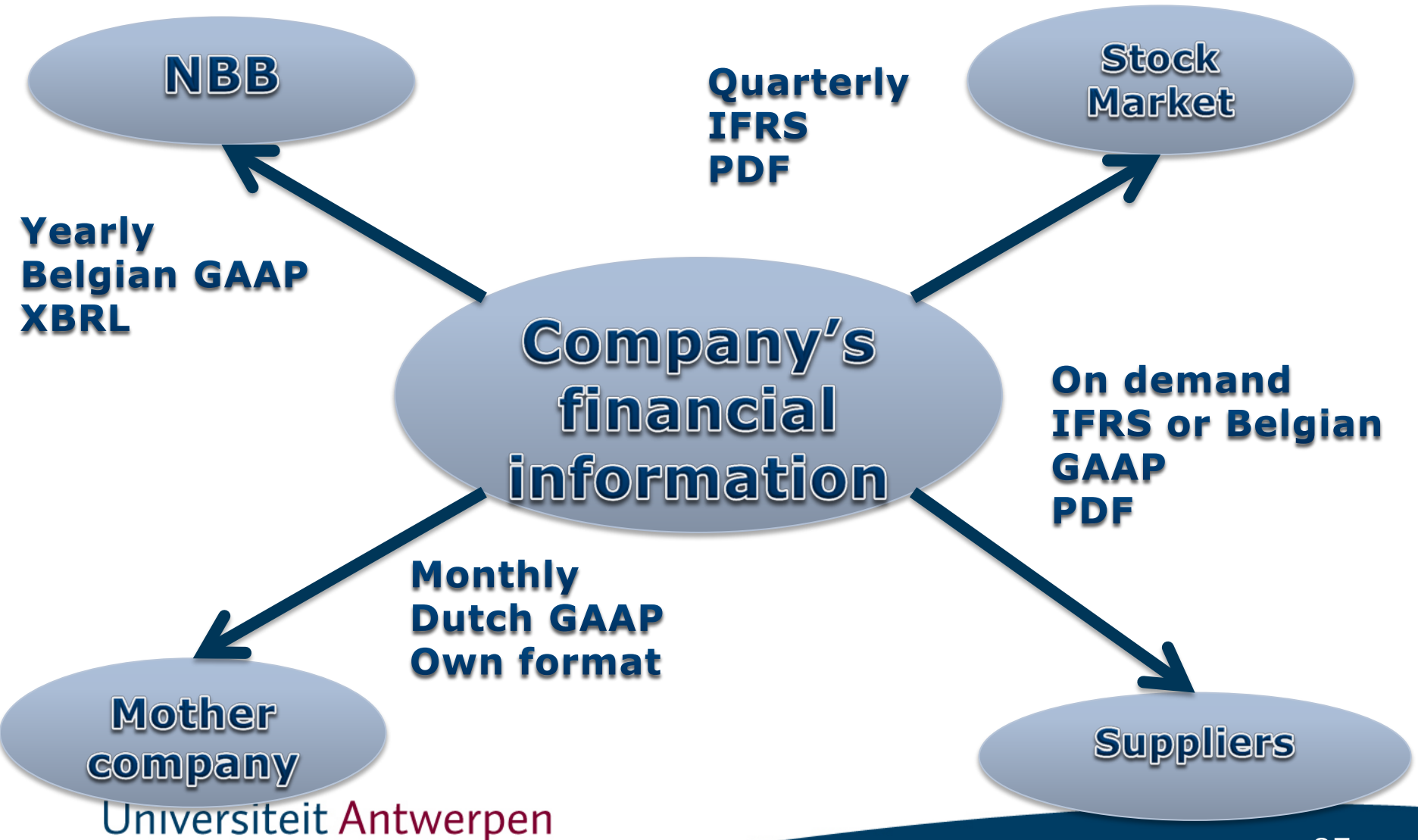
FREEMAN	ENGR 121	4cr	ENGR 100	ENGR 101	ENGR 102	ENGR 103	ENGR 104	ENGR 105	ENGR 106	ENGR 107	ENGR 108	ENGR 109	ENGR 110	ENGR 111	ENGR 112	ENGR 113	ENGR 114	ENGR 115	ENGR 116	ENGR 117	ENGR 118	ENGR 119	ENGR 120	ENGR 121	ENGR 122	ENGR 123	ENGR 124	ENGR 125	ENGR 126	ENGR 127	ENGR 128	ENGR 129	ENGR 130	ENGR 131	ENGR 132	ENGR 133	ENGR 134	ENGR 135	ENGR 136	ENGR 137	ENGR 138	ENGR 139	ENGR 140	ENGR 141	ENGR 142	ENGR 143	ENGR 144	ENGR 145	ENGR 146	ENGR 147	ENGR 148	ENGR 149	ENGR 150	ENGR 151	ENGR 152	ENGR 153	ENGR 154	ENGR 155	ENGR 156	ENGR 157	ENGR 158	ENGR 159	ENGR 160	ENGR 161	ENGR 162	ENGR 163	ENGR 164	ENGR 165	ENGR 166	ENGR 167	ENGR 168	ENGR 169	ENGR 170	ENGR 171	ENGR 172	ENGR 173	ENGR 174	ENGR 175	ENGR 176	ENGR 177	ENGR 178	ENGR 179	ENGR 180	ENGR 181	ENGR 182	ENGR 183	ENGR 184	ENGR 185	ENGR 186	ENGR 187	ENGR 188	ENGR 189	ENGR 190	ENGR 191	ENGR 192	ENGR 193	ENGR 194	ENGR 195	ENGR 196	ENGR 197	ENGR 198	ENGR 199	ENGR 200
---------	----------	-----	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------

- 250 variants of BP





# PhD Els Vanhoof (2015)





# PhD Dieter Van Nuffel (2011)

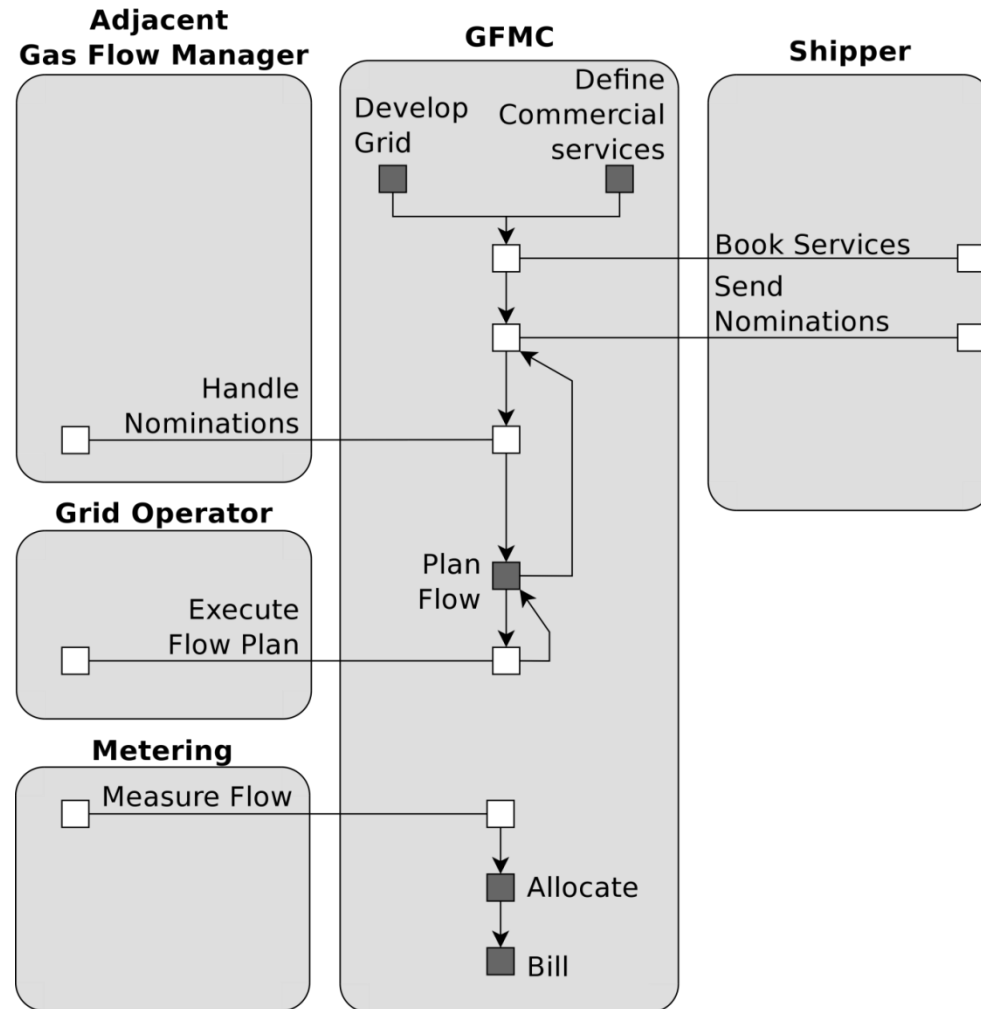
Case Studies

	Custom Bikes	Library	Public Service	Software Development	Training Institute	Purchase-to-Pay
Elementary Business Process	X	X	X	X	X	X
Elementary Life Cycle Information Object	X	X	X	X	X	X
Aggregated Business Process			X	X		X
Aggregation Level	X	X	X	X	X	X
Value Chain Phase	X	X				X
Attribute Update Request						X
Actor Business Process Responsibility			X			
Notification	X	X	X	X	X	X
Payment		X			X	X
Product Type	X		X	X	X	
Stakeholder Type			X			
Elementary Task	X	X	X	X	X	X
CRUD Task	X	X	X	X	X	X
Manual Task	X	X	X	X	X	
Managing Time Constraint Task	X		X		X	
Business Rule Task	X	X			X	
Bridge Task	X	X	X	X	X	X
Synchronization Task		X	X	X	X	
Synchronizing Task	X	X	X	X	X	X
Task Actor Responsibility				X		

- ...transposes NS principles to the level of Business Process Modeling, resulting in 27 principles

Table 6.1: Guideline Occurrences within the different case studies

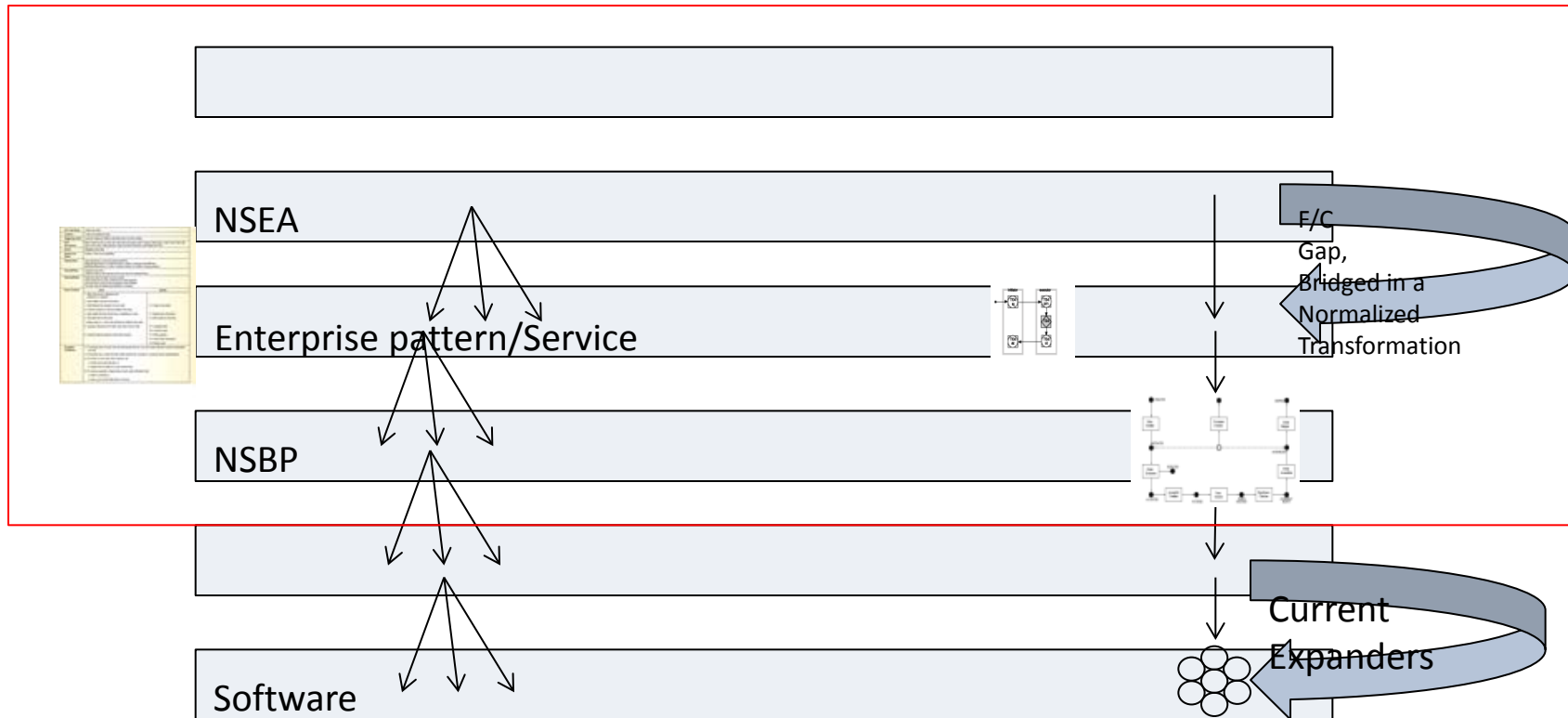
# PhD Philip Huysmans (2011)





# The Agile Organization

## Enterprise Engineering



Class Name
- attribute
- attribute
+ operation
+ operation
+ operation

SW constructen





# The Agile Organization

- Services Sector
  - "Door modularisering van dienstenproducten, standaardisatie van inputs en de stroomlijning van het voortbrengingsproces valt veel productiviteitswinst te behalen. De combinatie van ICT met niet-technologische innovaties vormt hiervoor de sleutel." (Van Ark, 2004)
  - "Information about the package is just as important as the package itself." (Fred Smith, FedEx)
- Industrial Sector
  - Ubiquitous Computing will increase the amount of ICT in industrial products.
  - ICT captures knowledge about the usage of products by customers, and the efficiency of the production process.
- "IT is not just transformational. It is also a central determinant of the successful business models and industry structure of a growing fraction of the economy." (Dhar et al., 2007)



HOME > BIZ > EXPERTS > MENINGEN

## Survival of the fittest

28/09/2013 om 15:30 door Peter Hinssen

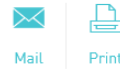


Foto: EPA

**Gewoon een wagen? Internetpionier Peter Hinssen stelt wat er zo speciaal is aan de elektrische Tesla.**

Vandaag is de laatste dag van onze Oracle Open Innovation Tour aangebroken. We rijden aan de overkant van de baai naar Fremont, waar de assemblage-fabriek van Tesla staat. De groep is 'pretty excited', want iedereen is benieuwd naar de hypermoderne fabriek waar het neusje van de zaai van de elektrische wagens wordt gebouwd. En iedereen wil er ook mee gaan testrijden natuurlijk. Als ik met 25 Belgische en Nederlandse CIO's de Tesla-fabriek binnenstap lijkt het alsof ik hen allemaal een golden ticket heb gegeven voor de chocoladefabriek van Willy Wonka.

En we worden niet teleurgesteld. De technologie van de wagens is verbluffend, de assemblage indrukwekkend, en vooral de testrit is amazing. Iedereen is verkocht. Maar het mooie aan Tesla is niet alleen de wagens, maar ook het verhaal erachter. Het verhaal van Elon Musk, de serie-ondernemer, die met Tesla de automobieliindustrie op zijn kop zet en een totaal nieuw merk lanceert dat je makkelijk de Apple-computer onder de wagens kan

mooie aan Tesla is niet alleen de wagens, maar ook het verhaal erachter. Het verhaal van Elon Musk, de serie-ondernemer, die met Tesla de automobieliindustrie op zijn kop zet en een totaal nieuw merk lanceert dat je makkelijk de Apple-computer onder de wagens kan noemen.

Maar Elon Musk is ook de oprichter van Paypal, dat de financiële sector aanpakte en een wereldspeler in online betalingen is geworden. En Musk is ook de oprichter van SpaceX, dat vorig jaar een contract met de NASA in de wacht sleepte van miljarden dollars om de Space Shuttle te vervangen. Musk wordt aanzien als de nieuwe Steve Jobs. Charismatisch, visionair, en vooral constant aan het kijken hoe je een industrie totaal kan omkeren. Musk is de grootste vijand van het status quo.

Eén van de deelnemers aan de Open Innovation Tour is Rudi Peeters, de CIO van KBC, dat één van de grootste IT-afdelingen in ons land heeft. Hij was degenen die Tesla in de proefrit tot het uiterste liet gaan. Ik vroeg hem wat hij onthoudt van dit weekje San Francisco.

"Om daar een kort antwoord op te geven: de schizofrenie van San Francisco en Silicon Valley. 23.000 startups die allemaal het nieuwe Google willen worden, maar ook een leger van daklozen die je overal op straat ziet. Life is a jungle here.

Laat het me verduidelijken. Er zijn twee fundamentele 'rules of nature' die hier spelen: Survival of the fittest, en het spel van Entropie.

Survival of the fittest voel je hier overal. Het gaat er niet om de grootste, snelste, slimste of goedkoopste te zijn, het gaat er wel om de meest flexibele, de meest agiele speler te zijn in het spel. Niet vastroesten. Flexibiliteit is key. De CEO van Tesla is daar een prachtig voorbeeld van: van Paypal naar een gloednieuw autobedrijf. Flexibiliteit geeft kansen op entrepreneurship en innovatie. Maar het geldt ook omgekeerd: bedrijven die niet flexibel en wendbaar zijn gaan er hier resoluut uit. Blackberry en Nokia zijn daar helaas prachtige voorbeelden van.

Maar de tweede regel is de regel van Entropie. In de natuur is elke crisis een opportuniteit. Een biotoop die verstoord geraakt zal die energie gebruiken om zich aan te passen. Een systeem dat in rust is, gaat uiteindelijk dood aan zijn Entropie. Je hebt dus druk nodig, een crisis, een verstoring. Als deze verstoring groot genoeg is dan kan het resultaat bijzonder verrassend zijn, dinosaurussen die verdwijnen, mutanten die plots betere kansen krijgen...

Natuurlijk zijn deze regels ook van toepassing voor bedrijven. Als je bij een bank werkt, ben je recent gebombardeerd tot ervaringsdeskundige inzake crisissen. In mijn sector spreken we dan over Too Big To Fail. Ik heb geleerd dat we wellicht beter spreken over Too Big To Change. Als we in onze biotoop vastgeroest raken, te rigide worden, dan is er maar een einde: failure.

De disrupties die we deze week gezien hebben hier in Silicon Valley, gaan niet over technologie, maar over de nieuwe businessmodellen. Het is aan ons om in ons eigen bedrijf een biotoop te creëren waarin onze goede talenten flexibel kunnen zijn, durven experimenteren, en ondernemen. Een cultuur van fail-forward bouwen, onszelf durven in vraag te stellen, en als bedrijf ons totaal durven hervormen. Als we blijven verder denken in de oude patronen zullen we verdwijnen. Het klinkt wellicht raar uit de mond van een bankier maar ik ben me er hier erg bewust van geworden: "Lang leve de crisis".



# The Agile Organization is the core of MIS (Beleidsinformatica)

- MIS is not about programming, but about using ICT to realize the Agile Organization
- Spectrum ranges from, and integrates:
  - Behavioral science (e.g., IT Management)
  - Design science (e.g., NS)



# Overview

- Inleiding
- Normalized Systems Theorie
  - Fundamenten: stabiliteit en entropie
  - Normalized Systems en software
  - Normalized Systems en bedrijfsprocessen
    - Enterprise Engineering
- **Alliantie voor Open Innovatie**
  - Technologie en innovatie
  - Normalized Systems Alliance



# Our Adoption Dream

**Change the IT world**

Replace Lehman by McIlroy



# Innovation and Adoption

- Many issues:
  - Gap between academia and business
  - Resistance to creative destruction
- Possible solution:
  - Network Enabled Collaboration
  - The Architecture of Participation
  - ➔ Reach out across boundaries



# Our Strategic Goal

Their achievement was sufficiently unprecedented to attract an enduring group of adherents away from competing modes of scientific activity. Simultaneously, it was sufficiently open ended to leave all sorts of problems for the redefined group of practitioners to resolve. Achievements that share these two characteristics, I shall refer to as "paradigms".

*Thomas Kuhn, The Structure of Scientific Revolutions.*



# Overview

- Inleiding
- Normalized Systems Theorie
  - Fundamenten: stabiliteit en entropie
  - Normalized Systems en software
  - Normalized Systems en bedrijfsprocessen
    - Enterprise Engineering
- **Alliantie voor Open Innovatie**
  - Technologie en innovatie
  - **Normalized Systems Alliance**





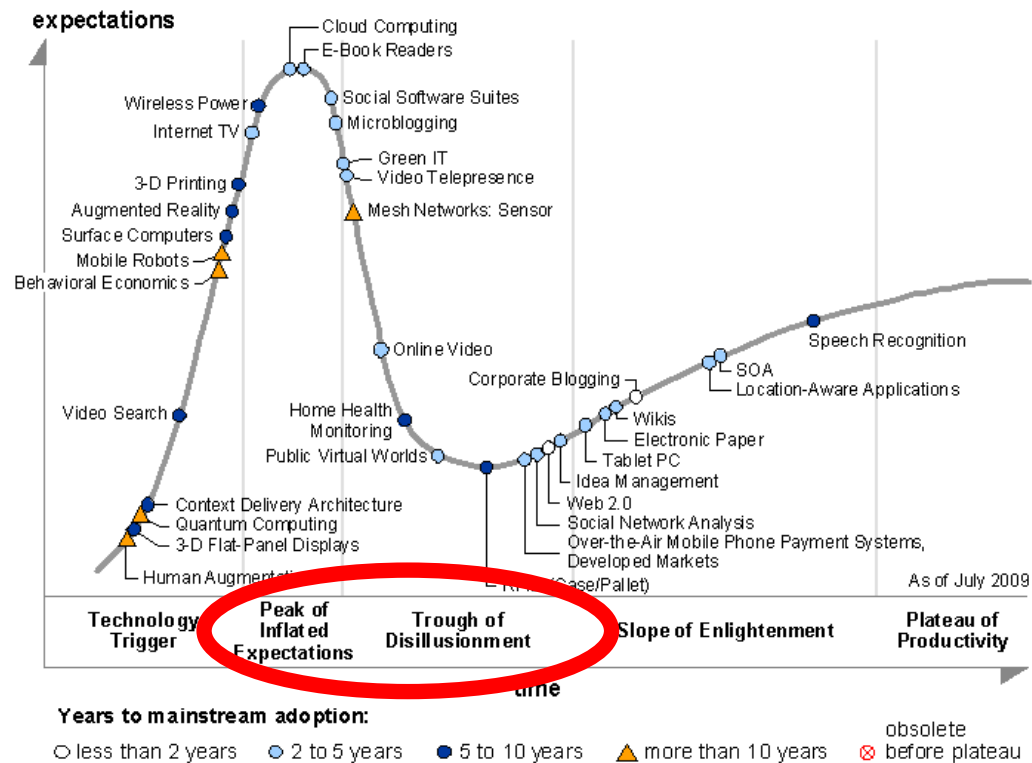
# Our Adoption Dream

**Change the IT world**

Replace Lehman by McIlroy

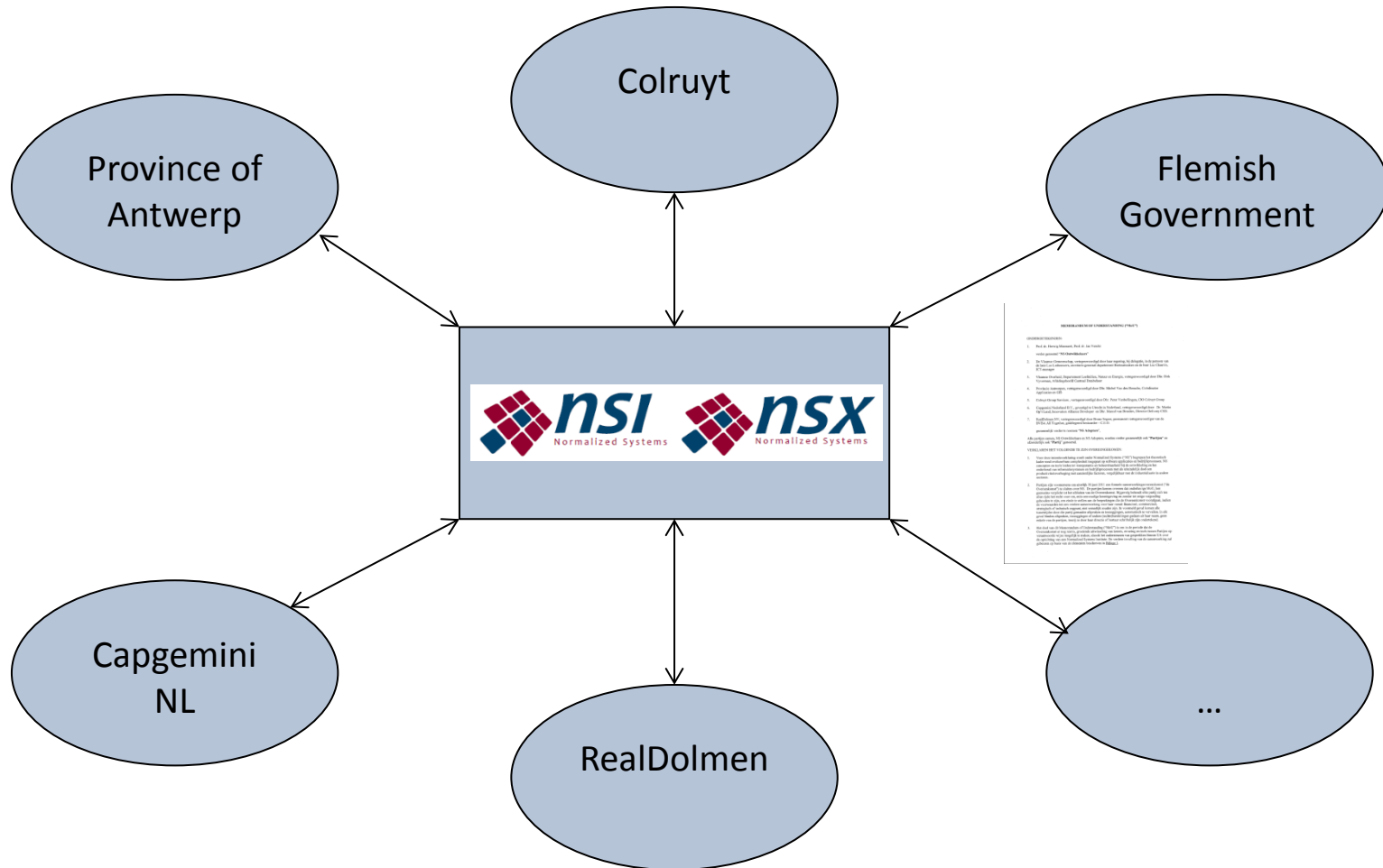


# The Adoption Reality





# NSA – 17 May 2011





# NSA – 17 May 2011

-  **nsi**  
Normalized Systems
  - Type IV institute of the University of Antwerp, affiliated with the Faculty of Applied Economics
-  **nsx**  
Normalized Systems
  - Official spin-off of the University of Antwerp
  - Development of the NS expanders
- Together, they are an ***academic spin-off 2.0***



# NSA – 17 May 2011

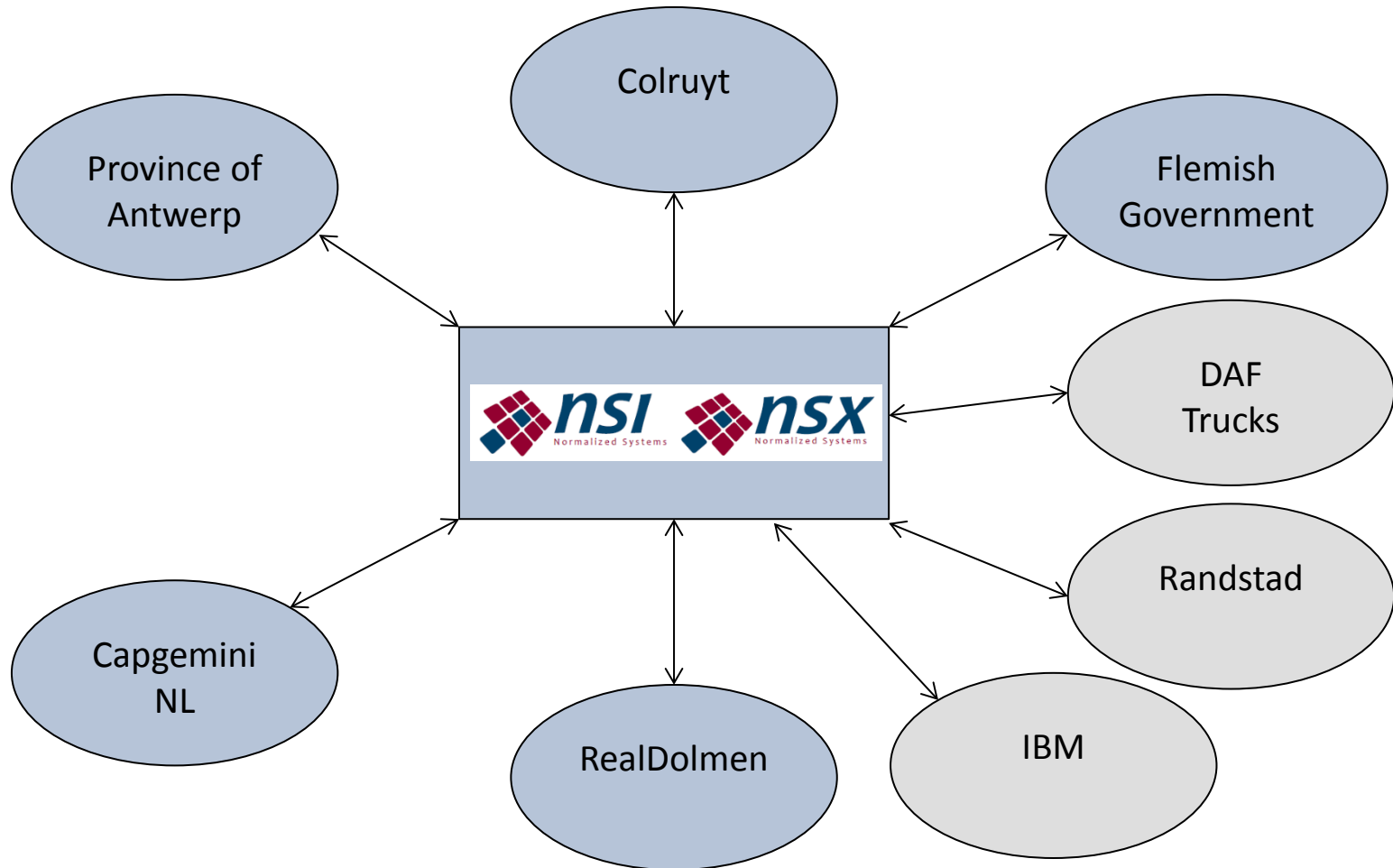
VERKLAREN HET VOLGENDE TE ZIJN OVEREENGEKOMEN:

1. Voor deze intentieverklaring wordt onder Normalized Systems (“NS”) begrepen het theoretisch kader rond evolueerbare complexiteit toegepast op software applicaties en bedrijfsprocessen. NS concepten en tools leiden tot transparantie en beheersbaarheid bij de ontwikkeling en het onderhoud van informatiesystemen en bedrijfsprocessen met als uiteindelijk doel een productiviteitsverhoging met aanzienlijke factoren, vergelijkbaar met de industrialisatie in andere sectoren.





# NSA – October 2013





# NSUA

- Computer Science Faculties
  - Prof. dr. Serge Demeyer (University of Antwerp)
  - Prof. dr. Jan Dietz (TU Delft)
  - Prof. dr. Alberto Silva, Prof. dr. Sergio Guerrero (TU Lisbon)
  - Prof. dr. Robert Pergl (TU Prague)
- Applied Economics Faculties
  - Els Vanhoof (University of Antwerp, dept. ACF)
  - Prof. dr. Chihiro Suematsu (Kyoto Univ, dept. Management)
- Research Centers
  - Centrum voor Wiskunde en Informatica
  - IBM Almaden Research



# Open alliance – our approach

- Full transparency to reduce risks
  - From research to industry: show expanders
  - From industry to research: audit applications
- MoU
  - Confirm industrial relevance of NS
  - Confirm the potential of a productivity factor
  - Share the investment cost & results
  - Establish NSI and NSX





# Open alliance – our approach

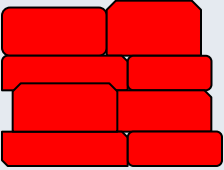
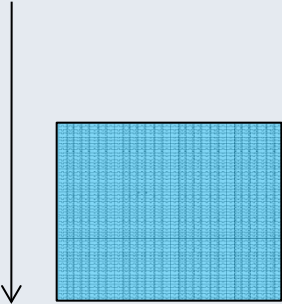
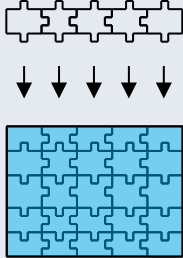

- Private business cases
  - All NSA partners
- Public business cases
  - Capgemini paper (2011)
    - Productivity factor 2-2.5 measured
  - HICSS paper 2014 (top 5 conference)
    - E.g. case from Province of Antwerp



# Conclusion

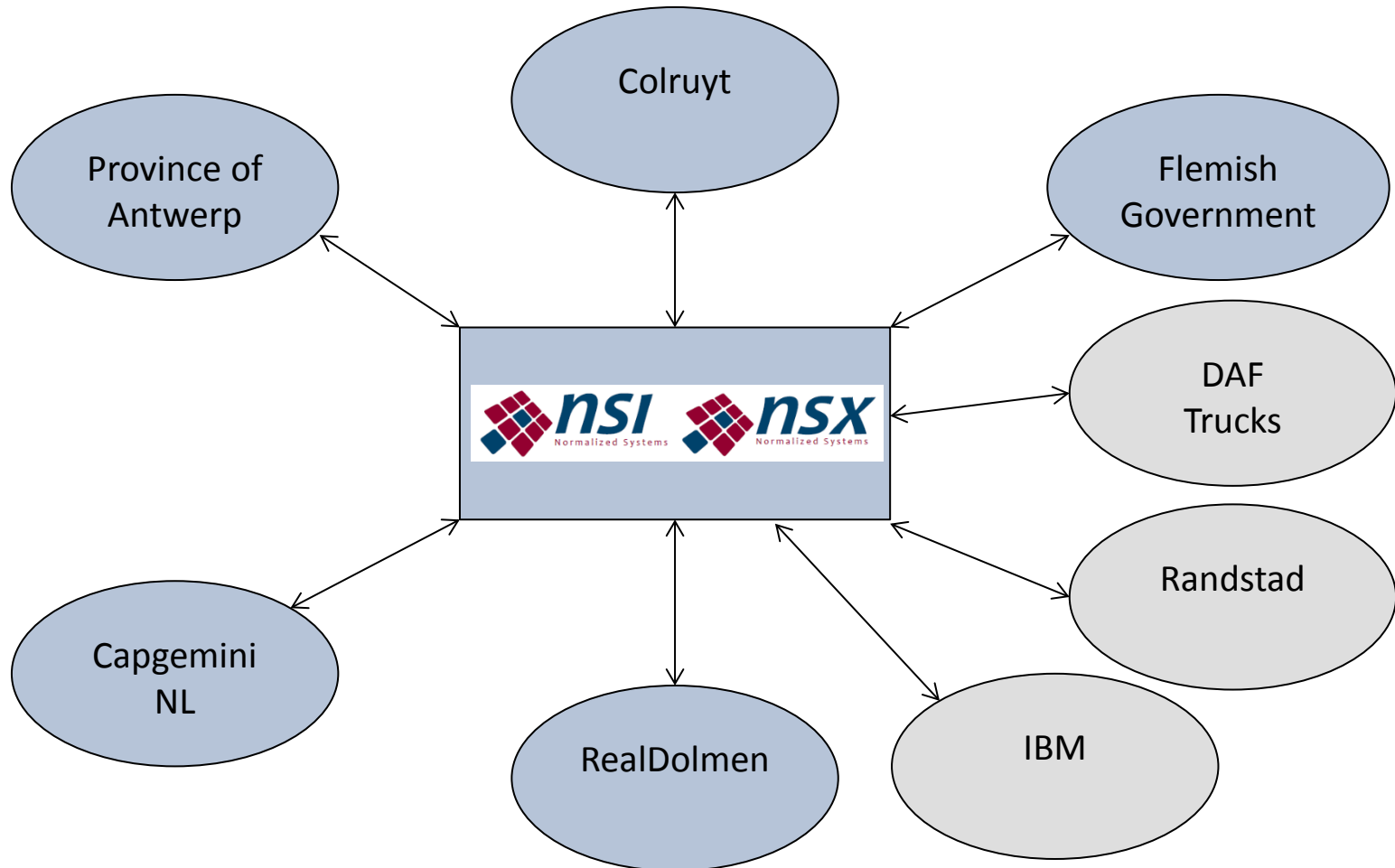


# NS Theory

Current	Step 1: Principles	Step 2: Elements	Step 3: Meet-in- The-Middle
<p>Lehman</p> 	<p>Fine-grained</p> 	<p>Expansion</p> 	<p>Aggregate <b>Reusable</b> And <b>Evolvable</b> Building Blocks → <b>DETERMINISM !</b></p> 



# NSA – October 2013





# Yearly updates

“Avond rond Beleidsinformatica”

14 november 2013



# Short Bibliography

- Mannaert Herwig , [Verelst Jan](#), [Ven Kris](#). - The transformation of requirements into software primitives : studying evolvability based on systems theoretic stability Science of computer programming - ISSN 0167-6423 - 76:12(2011), p. 1210-1222  
<http://dx.doi.org/doi:10.1016/j.scico.2010.11.009>  
<http://hdl.handle.net/10067/911120151162165141>  
[c:irua:91112]
  
- Mannaert Herwig , [Verelst Jan](#), [Ven Kris](#). - Towards evolvable software architectures based on systems theoretic stability Software practice and experience - ISSN 0038-0644 - 42:1(2012), p. 89-116  
<http://dx.doi.org/doi:10.1002/spe.1051>  
<http://hdl.handle.net/10067/934050151162165141>  
[c:irua:93405]
  
- [Huysmans Philip](#), [Oorts Gilles](#), [De Bruyn Peter](#), Mannaert Herwig , [Verelst Jan](#). - Positioning the normalized systems theory in a design theory framework Lecture notes in business information processing - ISSN 1865-1348 - 142(2013), p. 43-63  
<http://hdl.handle.net/10067/1089110151162165141>  
[c:irua:108911]



# Thank you for your attention!

Contact:

[herwig.mannaert@uantwerpen.be](mailto:herwig.mannaert@uantwerpen.be)

[jan.verelst@uantwerpen.be](mailto:jan.verelst@uantwerpen.be)

Universiteit Antwerpen