



Organizational Entropy as a Foundation for Enterprise Engineering

Prof. dr. Jan Verelst

Prof. dr. Herwig Mannaert

Universiteit Antwerpen





The Business Challenge

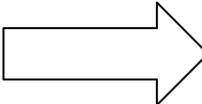
- The **Agile Organization**
 - Continually scans its ecosystem
 - Reacts quickly to opportunities and is innovative
- Has 2 Characteristics
 - **Increasing Complexity (Organized Complexity)**
 - Multi-channel vs. single channel
 - Diversify offerings/Additional services
 - **Increasing Change/Evolvability/Flexibility**
 - “These things are changing so fast it’s invention in the hands of the owner.” (Hansen et al., 2007)



Enterprise Engineering

Definition

“the whole **body of knowledge** regarding the development, implementation, and operational use of **enterprises**, as well as its **practical application**” (Dietz, 2006, p. 71).

- Current approaches **are**:
 - Heterogeneous
 - “the current literature on enterprise engineering consists merely of best practices, without an **integrating theory**” (Dietz, 2006, p. 71).
 - Descriptive
 - No “**normative** restriction of **design** freedom” (Hoogervorst, 2009)
 - Approaches **should be**:
 - Integrated
 - Prescriptive
- 



Entropy in Software Architectures

Universiteit Antwerpen

A dark blue decorative bar at the bottom of the slide, starting as a thin line on the left and thickening into a solid bar on the right.



The Dream: Doug Mc Ilroy



"expect families of routines to be constructed on *rational principles* so that families fit together as **building blocks**"

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.



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."
 - → CE are due to the way tasks, action entities and data entities are **combined or integrated !**

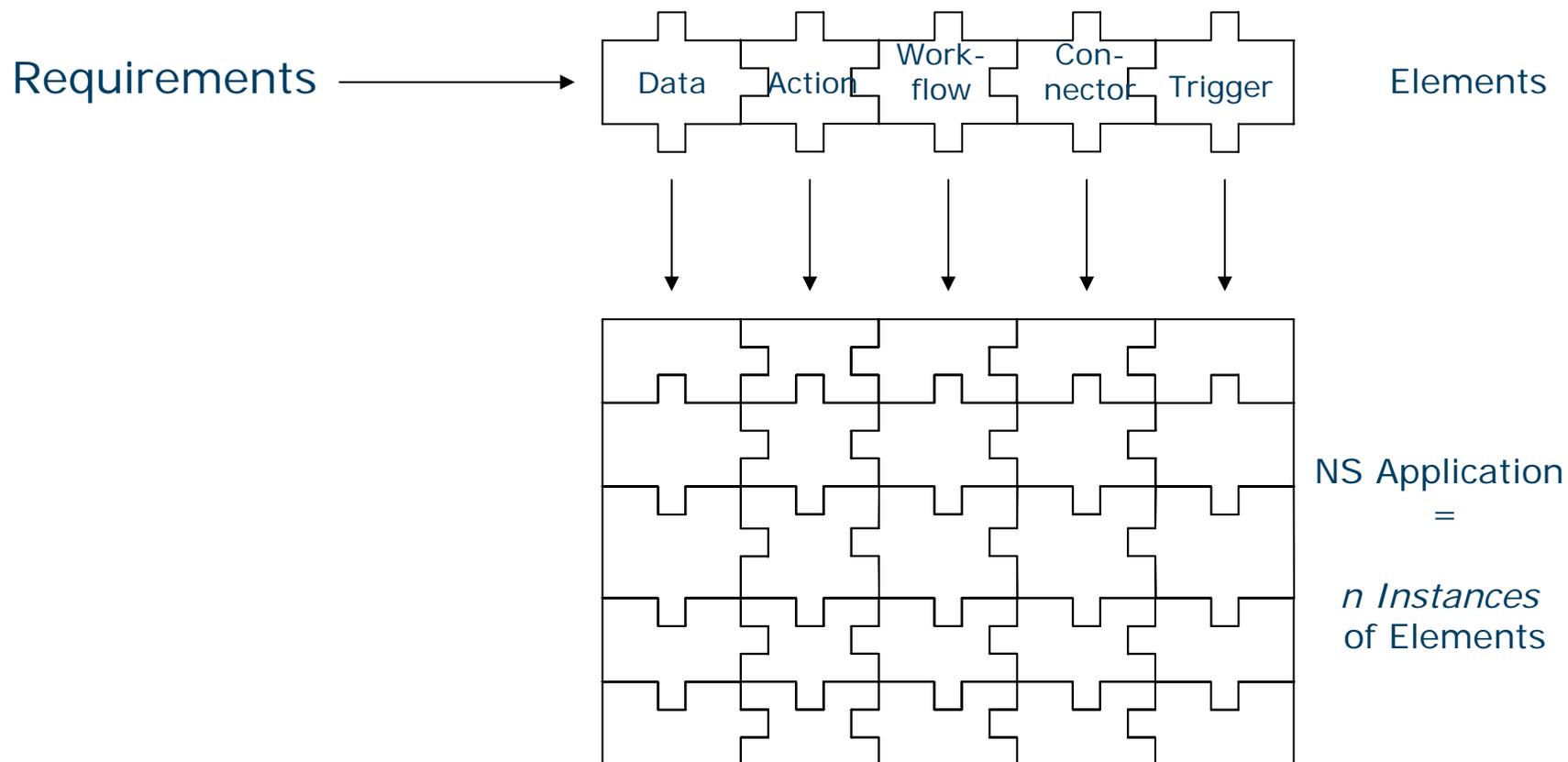


NS Principles and CE

- CE ...
 - Explain why McIlroy is incorrect
 - Which is *often* assumed correct by IT and non-IT people
 - Explain Lehman's Law of Increasing Complexity
 - CE suggest a view of information systems over time as ever increasing highly-coupled modular structures. **Highly coupled architectures** should therefore **not be considered coincidental, exceptional, or the result of substandard design**; all architectures automatically evolve towards them, and ever increasing preventive maintenance is required to avoid them.
- Conclusion
 - Omnipresent CE → No *evolvable* modularity !



Controlling CE using Elements





Elimination of CE → Determinism

- **Systematic elimination** of CE, using **fine-grained modular structures**, while controlling their inherent complexity
- → **Determinism**
 - All applications have similar fine-grained software architecture
 - product line or product factory
 - Impact analysis
 - Correctness
 - Reliability and Performance
 - Traceable execution
 - ...



Lehman Interpreted as Entropy

- Is an instance of the Second Law of Thermodynamics
 - ➔ Entropy !
 - “When given a system whose exact description is unknown, its entropy is defined as the **amount of information** needed to **exactly specify the state** of the system. This is because entropy represents the “potential for **disorder**” in a system.” (Wikipedia, 2009)
 - ➔ measure of “**what we do not know**”, or “a measure for our ‘**lack of knowledge**’ of a system”
 - ➔ **indeterminism** !



Entropy Interpreted

- Different descriptions exist
- Entropy in Statistical Thermodynamics
 - In Boltzman's definition, entropy is basically the number of possible **combinations** of individual **microstates** that yield the same **macrostate**. (Wikipedia, 2009)
 - → Entropy = **Knowledge of Combinations** (~ Integrations)
 - → Way to increase knowledge: increase **structure**
 - → If entropy is controlled → **Determinism** ! (We know!)
 - Examples
 - 100 coins, 50 heads, 50 tails
 - Without structure: #microstates= 10^{29}
 - With structure: 10 groups of 10 coins → #microstates = 2520
 - Gas container
 - With vs. without partitions



Round-up

- In Software
 - Modularity x Change → CE
 - **CE ~entropy**
 - They behave like Lehman: evolvability decreases, and preventive maintenance is necessary
 - If you eliminate them → **determinism** (and McIlroy!)



Organizational Entropy



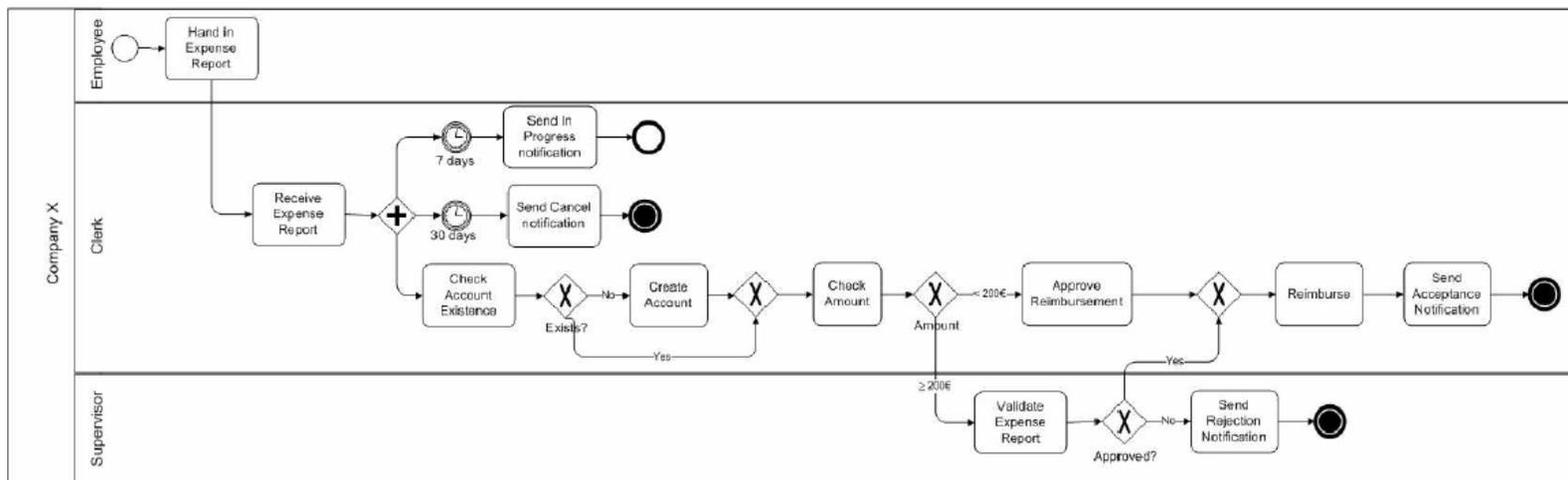
Modularity ?

- Dietz has rightfully warned us for technocratic thinking about enterprises, however:
 - A deterministic substrate remains necessary
 - And reduces complexity, by separating actually very different worlds (human & technology). Technology does not require communication solutions... enterprises do!
- 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, ...



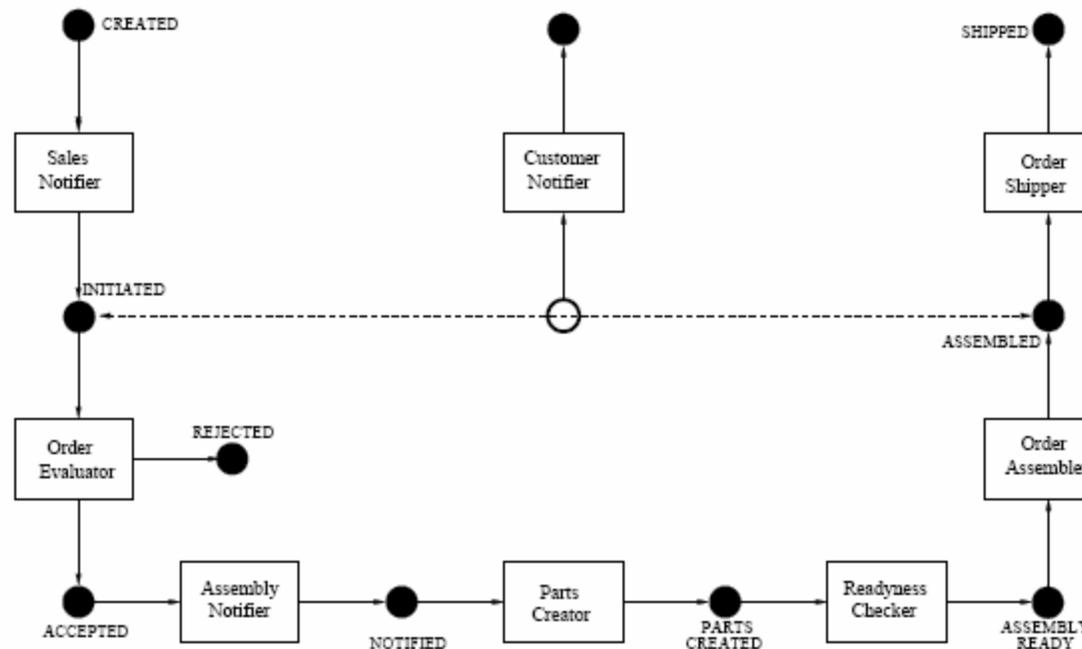
1. Entropy/CE at operational level

- Current Business Processes contain CE
 - → use NS principles to normalize BP into NSBP





State Machines





2. Entropy/CE at management level ?

- KPI's - typical scenario
 - KPI= increase customer satisfaction !
 - Operational level: yes, but how ?
 - → no correspondence between KPI's and operational decisions
- Interpretation of KPI's
 - KPI's = macrostates
 - A desired increase of a KPI, however, triggers the question which combinations of microstates can realize the macrostate
 - Currently, there are few if any design principles that link micro- and macro-level → entropy !



3. Current View on Enterprise Architectures

- Frameworks
 - Descriptive → evolvable ?
 - Few principles
 - Subjectivity
 - Limited systematic application of principles
 - Top down
 - View-oriented
 - Many modelling languages
 - Aimed at communication and early development ?
 - Limited traceability to code
- NS theory
 - Prescriptive → evolvable
 - NS principles
 - Normative restriction of design freedom
 - Extremely systematically applied
 - Bottom up
 - Aggregate elements
 - No modeling language yet
 - Aimed at system development
 - Traceability from NSBP to code
 - Need for EO !

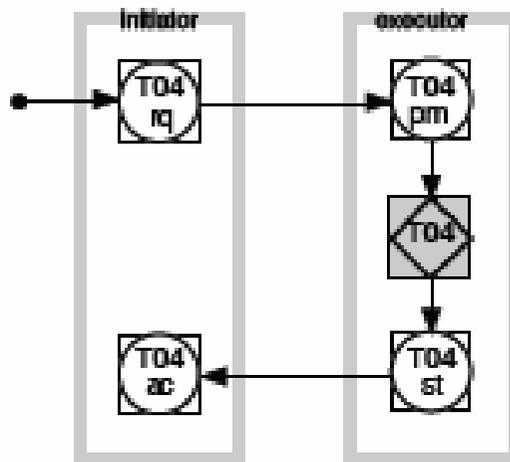


Entropy and DEMO/EO

- Explains fundamental observations in software engineering and enterprises
 - McIlroy
 - Lehman
- Next: Can entropy explain why **DEMO/EO** is so useful, as evidenced by e.g. the case studies in the PhD's of Hans Mulder and Martin Op't Land ?
 - Of course, DEMO/EO is based on LAP, but could DEMO/EO also be consistent with a highly technical concept such as entropy ?



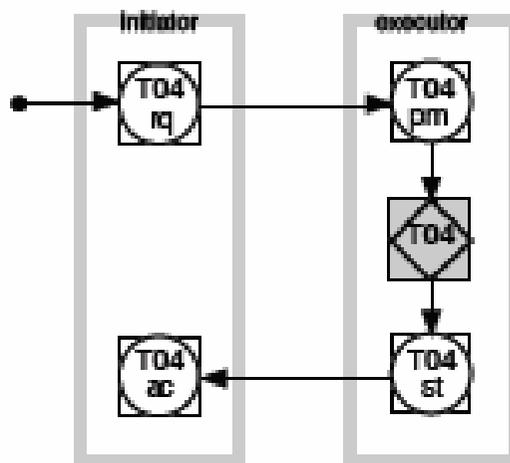
DEMO transactions



- The production act of a transaction seems to be a module consisting of a number of tasks, detailed in the action models.
- However, for each production act, there are 4 coordination acts → transactions are aimed at coordination-intensive problems, like enterprises consisting of human actors.
- Actually, such transactions define the interfaces of the modules !
 - Reminds of negotiation at operational level, but also project level (~IS acceptance problems)
 - Reminds of the partitions in the gas container
 - This is why DEMO/EO works so well: it is **human modularity**, which is used to control complexity, and...
 - → **controls entropy** → **determinism**
 - → design
 - → true Engineering !



DEMO transactions



- Reduce complexity by 70-90 %
- By using the transaction pattern, with the same internal structure, for all transactions
- = similar approach to elements, in controlling entropy !



DEMO Transactions

- Can Organizational Entropy be controlled (using Elements)?
 - Not very promising...
 - BPMN ?
 - Bpmn has GOTO, CE, mixes business and ICT level
 - Zachman
 - High-level, but imprecise
 - TOGAF
 - Organization-specific...
 - DEMO transactions are promising
 - Primary focus not on sequences of transactions, but on the individual transaction
 - Who does what, when, with which competences...
 - The transactions indeed have modular characteristics, and control entropy !
 - ~organizational construction rules (Op't Land ,2008)
 - Is high-level, but also precise !
 - CE ?



Future Research - Elements ?

- Suggest the following view on Enterprises at the **operational level**:
 - Controlling Entropy requires a fine-grained modular structure
 - While controlling its associated complexity
 - Element = +/- who does what when using which competences ?
 - Domain-specific and domain-independent elements **without CE !**
 - Describes transactions, products, services...
 - That can be instantiated
 - ~SOA: end user builds business processes
 - ~workflows: workflows have to be built for each individual instantiation
 - Aggregation = Bottom up perspective
 - Modular organizational structure
 - Modular business processes



Future Research - Elements ?

- Suggest the following view on Enterprises at the **management level**:
 - Several aggregated performance measures seem unproblematic
 - #claims processed...
 - However,
 - Is 'customer satisfaction' to 'operational' like functional and constructive ?
 - Are process patterns the answer ? The highly structured nature allows for better identification of causal relationships...
- Resulting in highly deterministic enterprises ?
 - Traceable execution
 - "the world is in great **need for transparency** about the operation of all the systems we daily work with, ranging from domestic appliances to the big societal institutions" (Dietz, 2006, p. 11)
 - Correctness...



Conclusion

- EE is searching for **prescriptive** and **integrating** theories
- At the software level, **entropy** has been shown to be:
 - Prescriptive: CE → evolvability decreases
 - Integrating: CE explain McIlroy, Lehman, ...
 - Systematically eliminating CE results in **determinism**
- We explored whether entropy at the enterprise level could be useful, and illustrated the use of:
 - Entropy at the operational level
 - Entropy at the management level
 - Entropy in evaluating current approaches to EE



Conclusion

- Based on our exploration, we propose to include entropy in the foundations of EE, as it would grant EE:
 - A prescriptive theory
 - Eliminating combinatorial effects, in order to reach **determinism**
 - An integrating theory
 - Providing business and ICT with a similar theoretical foundation
→ business/ICT alignment
 - Unifying higher-level and operational perspectives on enterprises
 - Addressing the essential challenges in the Agile Organization
⇔ e.g. Operations Research
 - A dynamic perspective
 - Dealing with change