CONTENT 2018. PATTERNS 2018. Joint Panel. Finding Essentials from Cloud-based Systems and Big Data. Namics.

BARCELONA, SPAIN, 22ND FEBRUARY 2018

Hans-Werner Sehring. Senior Solution Architect.

Agenda.

FIRST	Namics in a Nutshell.
SECOND	List of Panelists.
THIRD	Proposals.
FOURTH	Discussion.

FIRST

Namics in a nutshell.

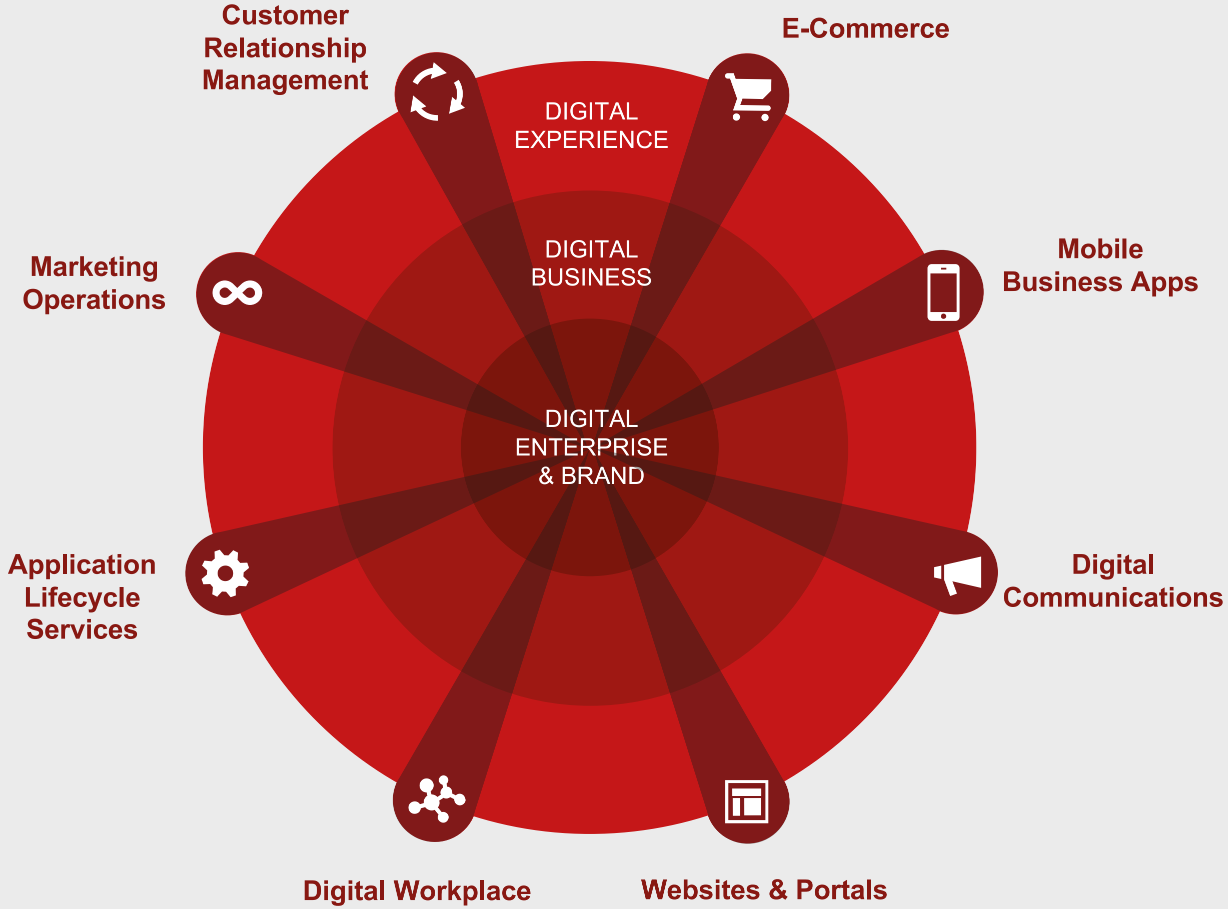
Creating winners of digital transformation

Bernd Schopp

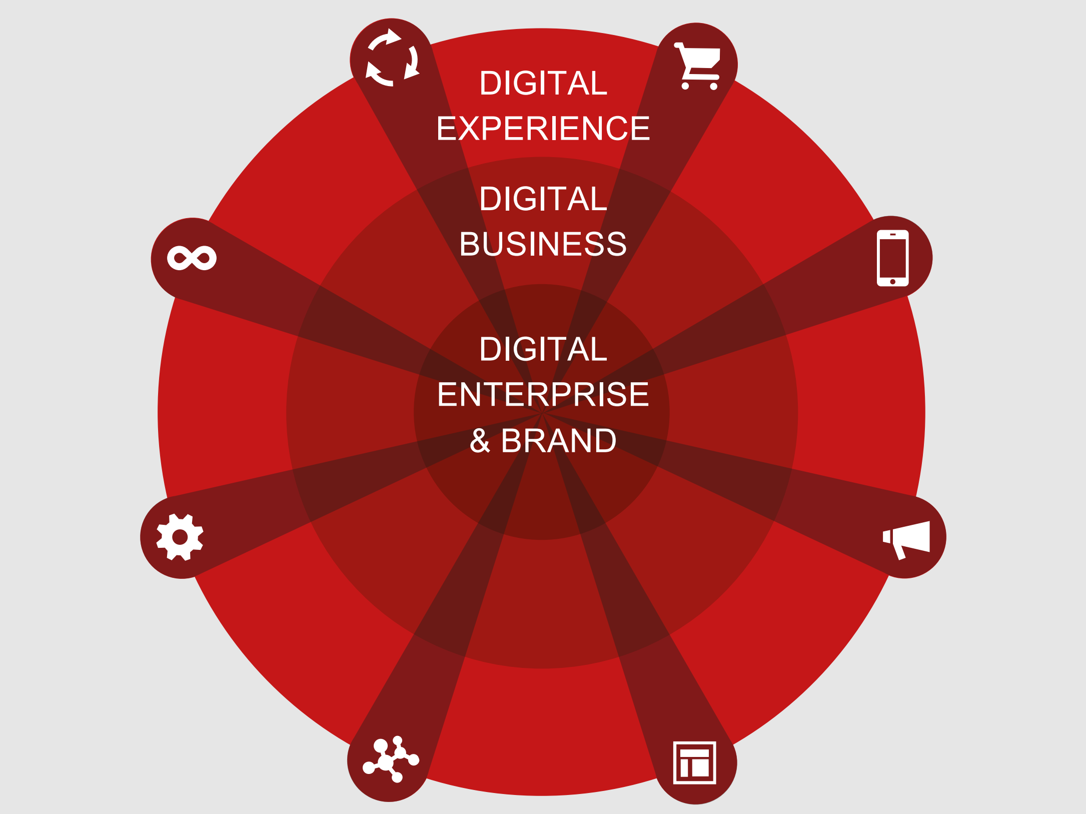
CEO, Namics

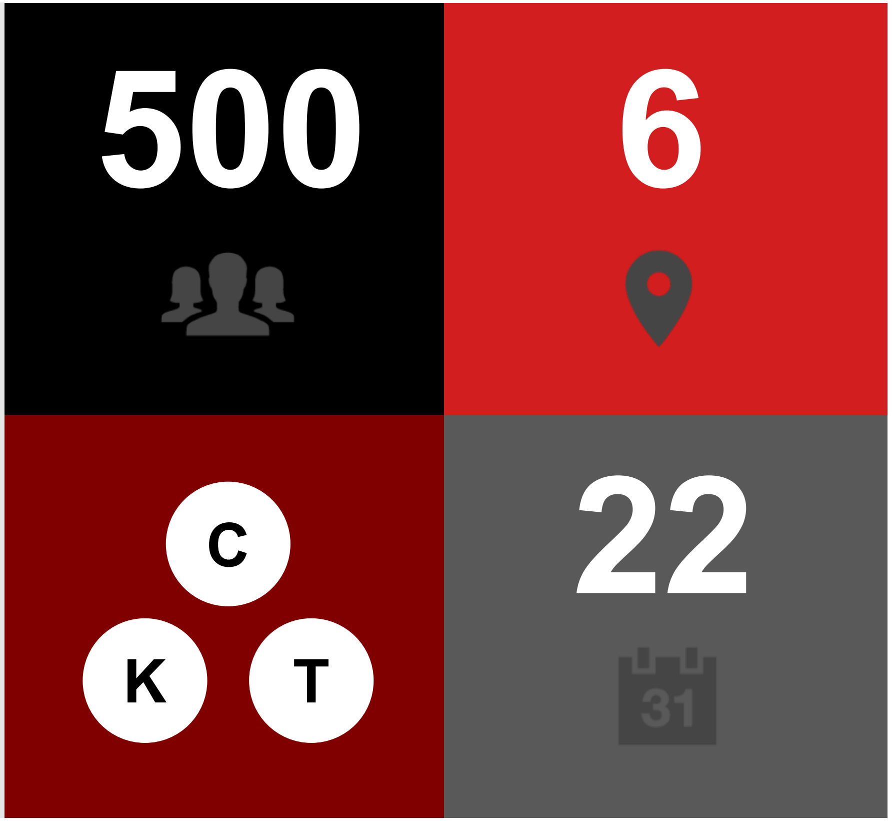
“With a commitment to high quality standards, we lead our customers from vision to reality and thereby enable market advantages and efficiency gains. We focus on the end-to-end user experience and based on this, the right measures on a strategic, technological and operative level.”

Our services.



Full Service. Close by. Namics.

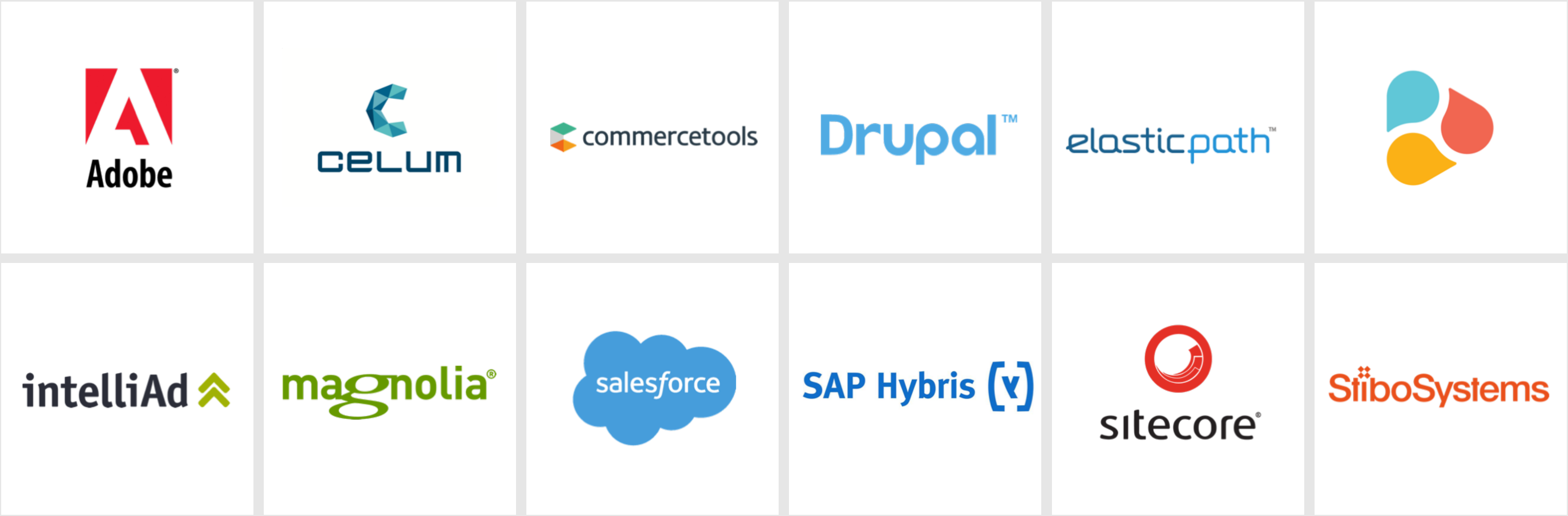




Owner-managed.

29 managing employees from Germany and Switzerland are partners of Namics.

Partnerships.



Award-winning

Commitment.

We have received several awards for our work, and our employees are very committed to sharing our industry knowledge and professional expertise. We are:

- **University partners**
- **Speakers at outside events**
- **Hosts of professional/industry events**
- **Authors of publications in trade media**
- **Writers of studies**

SECOND

Our Panelists.

Panelists.

- **Christian Kohls**
  TH Köln, Germany
- **Peter De Bruyn**
  University of Antwerp, Belgium
- **Alexander G. Mirnig**
  Center for Human-Computer Interaction, University of Salzburg, Austria
- **Hans-Werner Sehring**
  Namics

THIRD

Proposals.

Christian Kohls.

Alexander G. Mirnig.

Peter De Bruyn.

Hans-Werner Sehring.

Finding.

What do we look for?

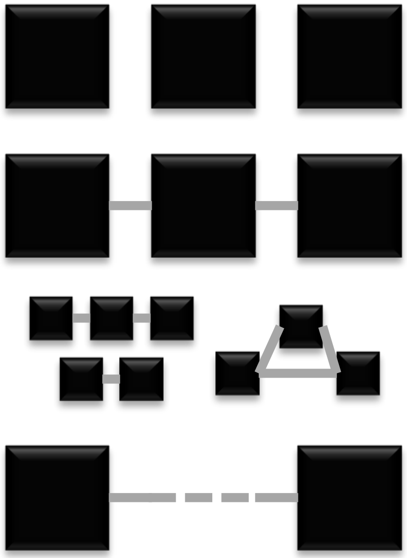
Big data processing relies on assumptions.

Data. We start with a hypothesis and search for evidence.
=> Querying.

Relationships. We suspect functional dependencies (denormalization).
=> Type checking?

Rules. Find repeating data constellations you do not know in advance.
=> Tractable?

Incomplete data, inconsistent data. We will have to deal with that.
=> Deductions, model checking, …



Essentials.

What is essential information?

Characterization of the meaningful in the pile of data.

This characterization is…

Domain-specific because interpretation of the same data varies. Ex.: customer journeys over commerce website from perspective of sales and support departments.

Application-specific because data depends on the representations chosen by the data-producing systems. Ex.: units of measurement.

From the Cloud-based Systems and Big Data.

Where is the data? Where is the software analyzing it?

Data shipping vs. function shipping was the classical question of the past. Big data will not be shipped for sure. So the software analyzing it is either a service in the (same) cloud or software deployed there.

How did the data get there? Incrementally, e.g., sensor data (IoT), behavioral data (tracking). This means, it is historical data.

How to process a data stream? Can it be processed as it comes (stream reasoning), or can we consider isolated time intervals (then it’s not actually big data), or do we have to process all data again and again (real-time CC)?

Divide and Conquer.

What can we actually offer for big data analysis?

There need to be sophisticated analysis services. But they are domain- and application specific.

Microservices makes sense in the Cloud.
E.g., Amazon Lambda, “Service as a Service”, FaaS.

Allow integrating domain- specific logic. We set our own goals and interpretations. IF there is some kind of framework to hook the code in.

Can be arranged in architectural styles as Pipes&Filters or Mediator. This way, they can lay the foundation for stream processing.

Towards an Infrastructure for Big Data Analysis in the Cloud.

A Cloud-based Framework.

For digesting (big) data pools. Based on the assumption this framework offers:

Base services for efficient (big) data analysis, checks, deductions.

Hooks for Microservices that can be hooked into the Cloud’s macro services for domain-specific goals and interpretations.
E.g., AWS Lambda, “Service as a Service”, FaaS.

Integration of common-sense knowledge for the particular domain.
This is the foundation for data interpretation and cleaning.

FOURTH

Discussion.

hans-werner.sehring@namics.com. Senior Solution Architect.

Thank you. Namics.

CONTENT. PATTERNS. Finding Essentials from Cloud-based Systems and Big Data. Panel. Namics.

BARCELONA, SPAIN, 22ND FEBRUARY 2018

Hans-Werner Sehring. Senior Solution Architect.

FIRST

Panel. Positions. Proposals.

Christian Kohls.

Patterns describe a meaningful solution structure.

Iterative process of …

- detecting patterns,
- writing them down in theory, and then
- offering them to others to apply/validate them again.

Alexander Mirnig.

Two things to be considered:

1. Constraints from various domains are connected by the application domain.
2. Changing context.

Peter De Bruyn (1/2).

Patterns for designing maintainable software.

Reusing existing solutions allows spending more time on a solution.

Recurring structure leads to complexity reduction.

Evolvability shows where to adapt things (changing the pattern you know you have change every location where you applied the pattern).

Peter De Bruyn (2/2).

Pattern challenges:

- requires time
- inductive process
- iterative process
- delayed pay-off
- many existing patterns are
  - underspecified
  - unknown
  - limited in adoption
- need for instantiation and collaboration mechanism

SECOND

Panel. Discussion. Conclusions.

Discussion (1/2).

Observation:

There are high- and low-level patterns.

Question/challenge:

Are there too many patterns? How to judge quality?

Patterns do take constraints from multiple domains into account. But they don't claim to be the best solution to a problem, only in a certain context.

Discussion (2/2).

Application in law:

Contracts, jurisdiction seem to work pattern-based (without knowing).

Value of patterns:

Patterns are a good starting point for learners. But do they hinder innovation? Disagreement on this point.

Value of patterns 2:

Open point: Can patterns serve as "natural laws" for software (that has no such laws)? This way describing how things relate?

THIRD

Outlook.

Further discussion.

Impressions:

- A lot to be discussed concerning meaning of patters.
- A lot of knowledge and experience gathered on the panel.
- Different views on value of pattern application.

Concluding question:

Should we have a tutorial or a workshop on patterns best practices next conference?

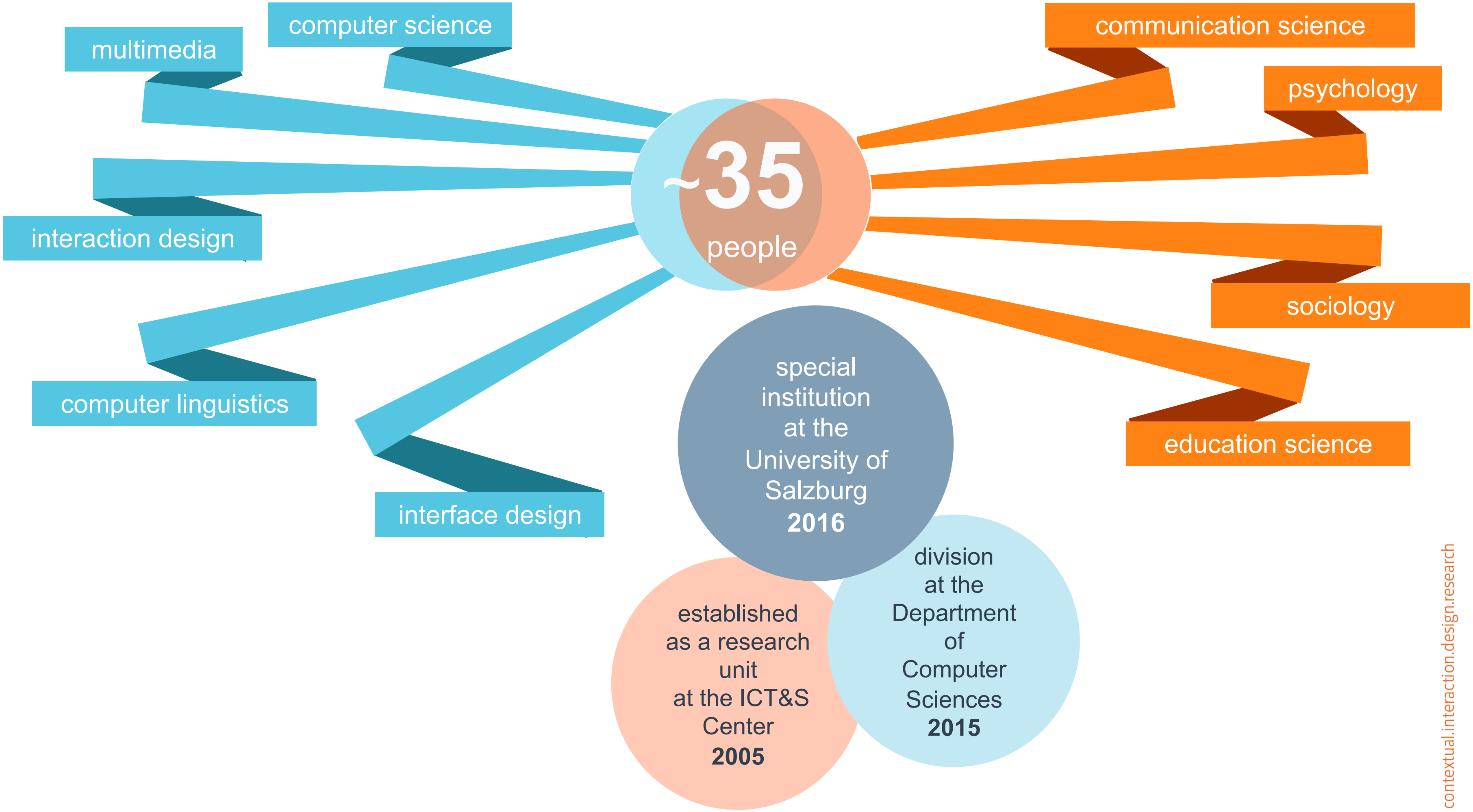
hans-werner.sehring@namics.com. Senior Solution Architect.

Thank you. Namics.

Patterns: About Not Reinventing the Wheel

On the difficulties of mining and reusing patterns in rapidly changing contexts

Background – Center for HCI in Salzburg



special institution at the University of Salzburg **2016**

established as a research unit at the ICT&S Center **2005**

division at the Department of Computer Sciences **2015**

contextual.interaction.design.research

Automotive Design

* © Arno Laminger 2015

contextual.interaction.design.research

Automotive design patterns

Capture solutions for in-vehicle designs

But: What constitutes successful in-vehicle design is influenced by a number of factors from both inside and outside the HCI discipline

- UI design best practices (color pairs/sets, menu depth, design for affordance, …)
- Distractive potential of interacting with a device (remember: driving is the primary task)
- Legal constraints (unobstructed view on the road, presence of standard gauges and signals, ...)
- ... and many others

Changing Context

Proven Solutions:

- A (design) pattern is a structured documentation to a **proven** solution to a reoccurring problem, embedded in at least one of the contexts it occurs in.

But: Technology can progress so rapidly, that working solutions have little time to become established before they are obsolete. The automotive domain is one such rapidly progressing domain.

contextual.interaction.design.research

Changing Context

The Proposition:

- When working solutions in one domain depend on best practices from other domains, these practices do not necessarily have the same status or relevance across said domains
- This means that working solutions might be obsolete in one context, but not in another
- It further means that a problem might be novel to one particular domain, but already have a solution somewhere else
- Extending pattern mining beyond context and discipline the problem occurs is necessary to not needlessly reinvent the (steering-)wheel.

contextual.interaction.design.research

Contact

Mag. Alexander G. Mirnig
Center for Human-Computer Interaction
University of Salzburg

Jakob-Haringer-Straße 8 / Techno 5
5020 Salzburg

+43 662 8044/4834
alexander.mirnig@sbg.ac.at

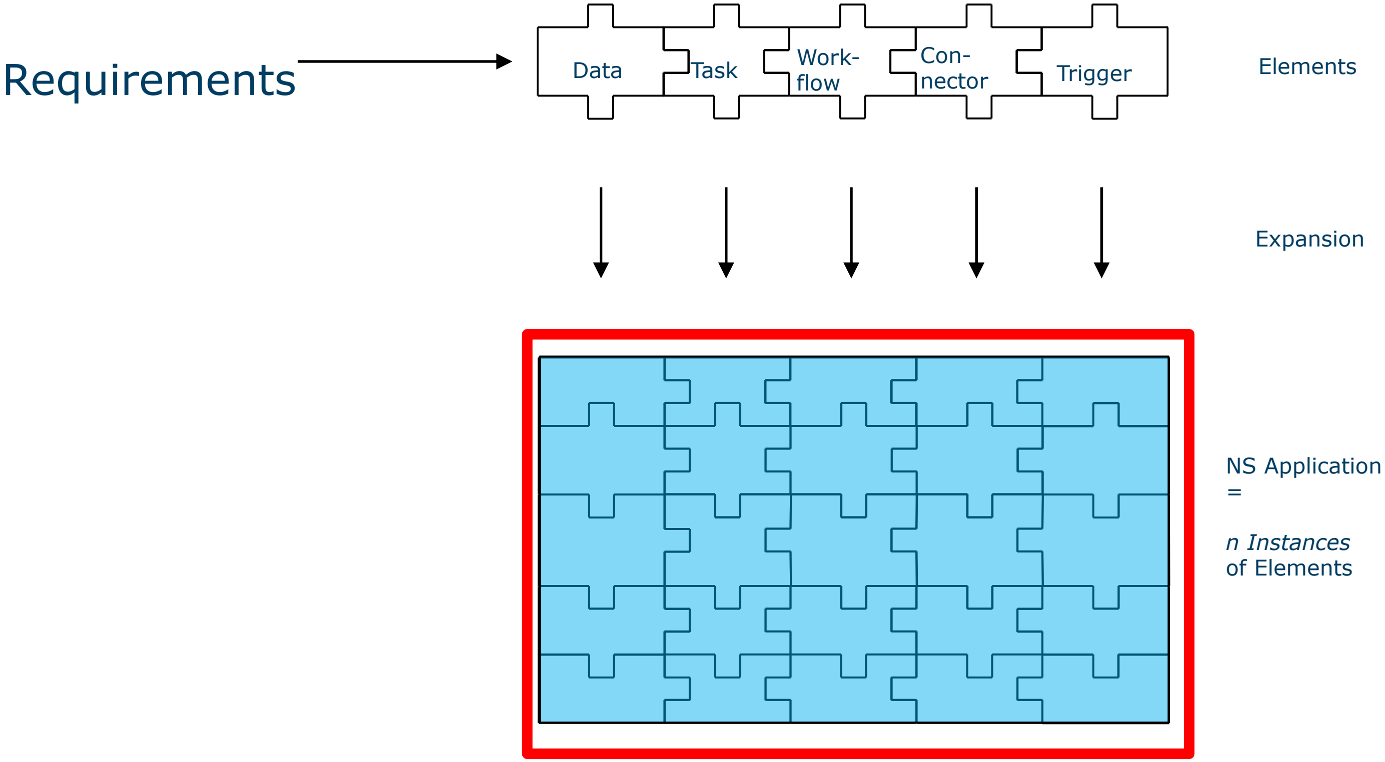
Panel discussion

CONTENT/PATTERNS 18

Peter De Bruyn, Ph.D.
Department of Management Information Systems
University of Antwerp

Universiteit
Antwerpen

Patterns in our research group – University of Antwerp



Requirements

Data

Task

Work-flow

Con-nector

Trigger

Elements

Expansion

NS Application
=

n Instances
of Elements

Patterns: why?

- **Reuse** of existing solutions
  - no reinvent the wheel
  - put much effort into one “very good solution” and then reuse it all over again
- Recurring structure allows **complexity reduction**
  - every pattern instantiation can be considered as a substructure
- **Evolvability**: you know where to adapt things

Patterns: challenges/questions/propositions

- **Requires time**
  - inductive process
  - iterative process
  - delayed pay-off
  - >> how to “buy” this time?

- Many **existing patterns** are
  - underspecified
  - unknown
  - >> how to “increase adoption”?

- Need for an **instantiation and collaboration mechanism**
  - >> how to enable this?