Self-Learning, Self-Actuating and Decentralized Control: How Emergent System Capabilities Change Software Development

SWESoS, Gothenburg, September 9th, 2016.

Helena Holmström Olsson, Malmö University
&
Jan Bosch, Chalmers University of Technology

Big Data

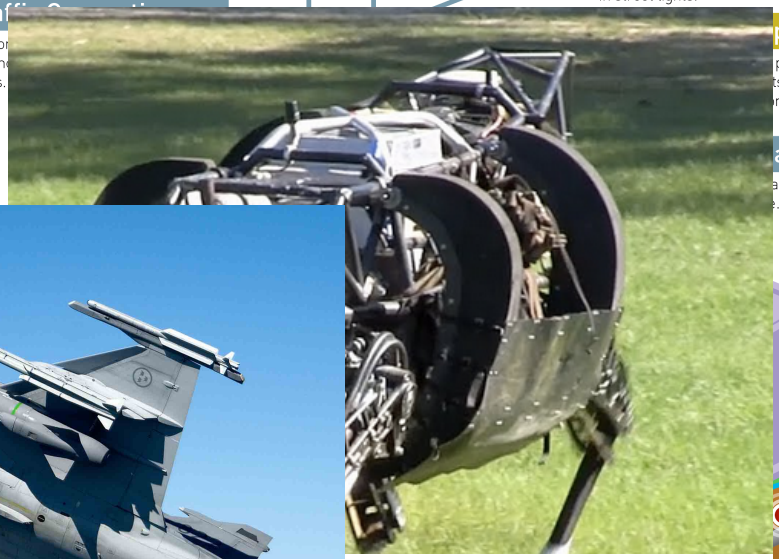
Air Pollution
Control of CO_2 emissions of factories, pollution emitted by cars and toxic gases generated in farms.

Smartphones Detection
Detect iPhone and Android devices and in general any device which works with Wifi or Bluetooth interfaces.

Electromagnetic Levels
Measurement of the energy radiated by cell stations and and WiFi routers.

Smart Roads
Warning messages and diversions according to climate conditions and unexpected events like accidents or traffic jams.

Smart Lighting
Intelligent and weather adaptive lighting in street lights.



Gripen Drone

Water Leakages
Detection of liquid presence outside tanks and pressure variations along pipes.

Vehicle Auto-diagnosis
Information collection from CanBus to send real time alarms to emergencies or provide advice to drivers.

Item Location
Search of individual items in big surfaces like warehouses or harbours.

Smart Parking
Monitoring of parking spaces availability in the city.

50 Terabytes of data are created every second

Background

- ***Connected*** systems allowing continuous data collection and experimentation (A/B testing).
- ***Interconnected*** systems that dynamically discover one another, and seamlessly interconnect during runtime.
- ***Intelligent and autonomous*** systems that adapt and improve over time and that proactively initiate actions.

Emergent System Characteristics

- ***Self-learning systems:*** Adaptive systems whose operation algorithm improves based on trial and error.
  - The system learns to ***reconfigure and adapt*** itself to new or changing inputs.
  - The system experiments with different behaviors and learn to more ***rapidly adjust their behaviors*** according to e.g. user preferences (A/B testing).
- ***Self-actuating systems:*** Systems that actively initiate actions based on input from the environment in which they operate.
  - The ***system initiates actions*** – not the user.
  - Sensing, actuation and control to analyze a situation and make informed ***decisions*** based on available data.
- ***Decentralized systems:*** Systems in which each master in the network has all data. This supports local decision-making and allows for rapid actions to be taken in the network.
  - Network nodes share data to support ***local decision-making*** and rapid actions to be taken in a decentralized manner.

Research Objectives

- We explore some of the ***emergent system characteristics*** that we believe pose new challenges on software development.
- We explore the ***transition*** that many software development companies are currently experiencing in relation to R&D practices.
- We present a development approach where software development is no longer only a human effort, but instead a ***synergy*** between human R&D teams and autonomous systems.

Research Method

- Case study research in three domains:
  - *Embedded systems*
  - *Online gaming*
  - *Internet of Things (IoT)*
- We conducted interview studies, group interviews, workshops, observations and validation sessions.
- Roles representing software development, release, project and product management and sales and marketing.

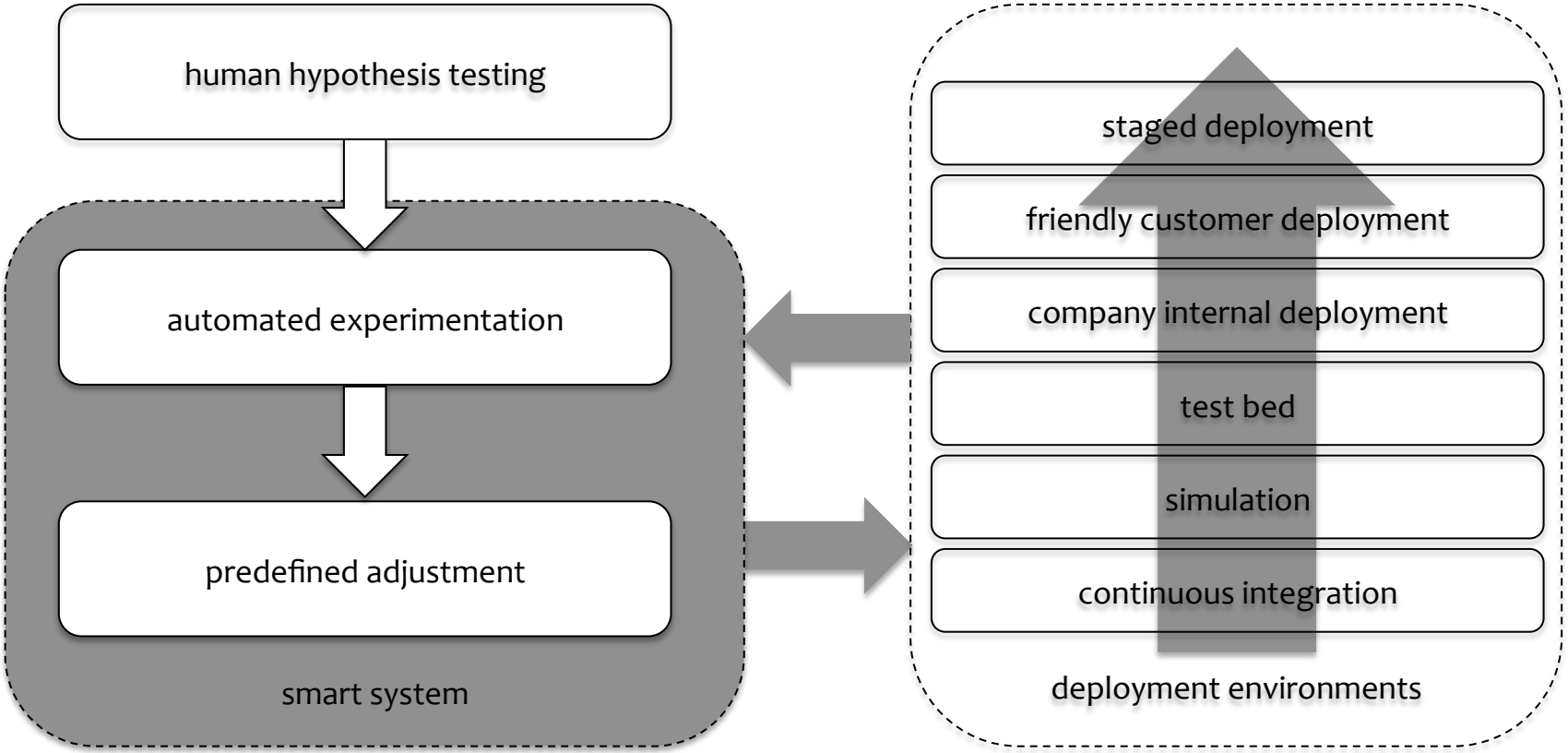
Software Development Practices

- All systems will employ ***continuous deployment*** (at least once per agile sprint).
- R&D teams will employ A/B testing for all feature development and "MVP" approach for new products (***instead of requirements!***).
- Systems will use streaming analytics in various forms and exhibit ***data-driven*** behaviors.
- Systems will ***autonomously*** experiment with their behavior to improve the delivered value.
- Families of similar systems will ***learn from each other*** automatically.

Towards Continuous Evolution of Smart Systems

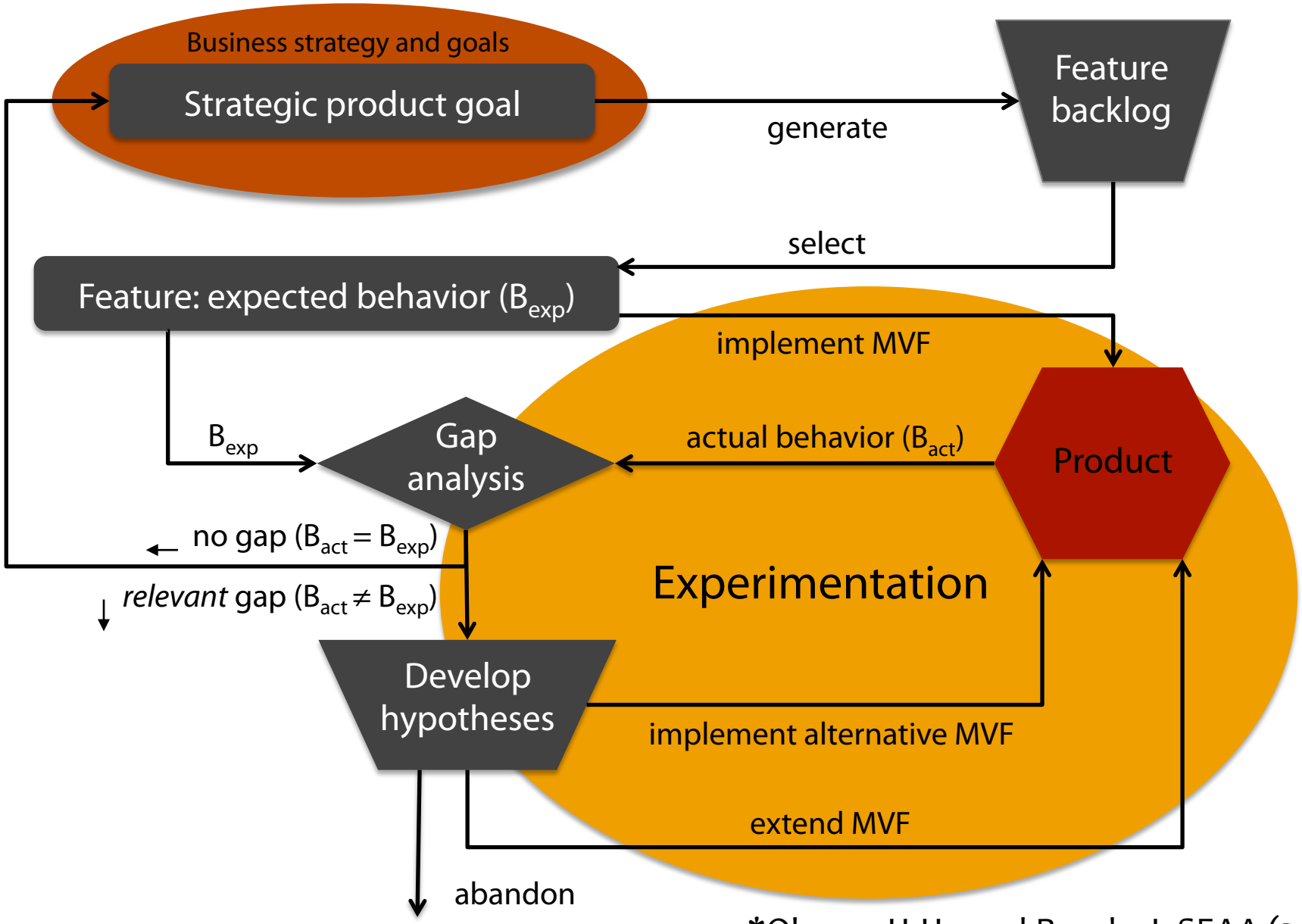
Characteristics	Traditional development	Agile development	Data-driven development	Continuous and experimental development
R&D process	• Long cycles with defined milestones • Waterfall and sequential	• Rapid cycles with short development sprints • Iterative and incremental	• Frequent and automated experimentation. • Data analytics • Dynamic and evolutionary	• Automatic experimentation by systems in the field • Systems exploring and optimizing alternatives
Data collection and use	• Pre-study • Requirements specifications	• Customer collaboration • Product owners and/or customer-specific teams as proxy	• Data collection from systems in the field • Continuous validation with customers	• Automated experimentation. • Embedded analytics in systems in the field
Business and organization	• Discipline oriented organization • Technology-driven innovation	• Cross-functional R&D teams • Customer-driven innovation	• Integrated R&D, PdM and data analytics team • Data-driven innovation	• R&D teams and smart systems • Synergy-driven innovation

Towards Continuous Evolution of Smart Systems*



*Bosch, J., and Olsson, H.H. SEAMS, SEAA, (2016)

Human Hypothesis Testing (HYPEX*)



*Olsson, H.H., and Bosch, J. SEAA (2013).

Automated Experimentation

- Automated experimentation takes place in deployed systems.
- The R&D team has provided functionality to the system to automatically experiment with different responses to the same stimuli.
- The system is configured to experiment with its behavior within predefined boundaries.
- The system randomly selects a response based on input data.
- The “success” is clearly defined and the system measures gap between realized outcome and optimum.
- With sufficient “depth” the system learns what the optimal response for each input data range is.

Predefined Adjustment

- Once the automated experimentation has concluded on an optimal set of responses for each range of input stimuli, the experimentation can be concluded and a set of standard responses can be statically defined.
- Using static, defined responses in combination with streaming analytics, the system selects a response.
- System response can be generic or personalized for each system and/or user.

Challenges: Elements for Automated Experimentation

- R&D teams need ***techniques*** to express the boundaries, the required type of experimentation and the value function that the experiment should optimize for.
- The system needs an ***architecture*** that is designed for data collection, automated experimentation and that can track its own performance.
- Solutions are required to share the ***learnings*** from all systems in a family such that all systems benefit from the experiment.

To summarize

- Emergent system characteristics pose new challenges on how ***to R&D*** these systems.
- We are moving towards ***automated*** experimentation by systems that over time learn from data and from other systems.
- Combining ***human and automated*** experimentation is an unsolved, but increasingly important problem.
- We present a ***high-level proposal*** to addressing this challenge.
- The approach has ***architectural and ways-of-working*** (R&D) implications.

Conclusion

- Self-learning, self-actuating and decentralized systems require a ***fundamental shift*** in software engineering practices.
- ***Human experimentation*** of software needs to be complemented with ***automated self-experimentation*** by families of systems.
- ***No software engineering approaches***, including architecture, processes and organizational dimensions, that addresses human/ autonomous/data-driven behaviors exist today.

Thank you!

helena.holmstrom.olsson@mah.se
jan.bosch@chalmers.se