Competence Centers for Excellent Technologies
www.ffg.at/comet

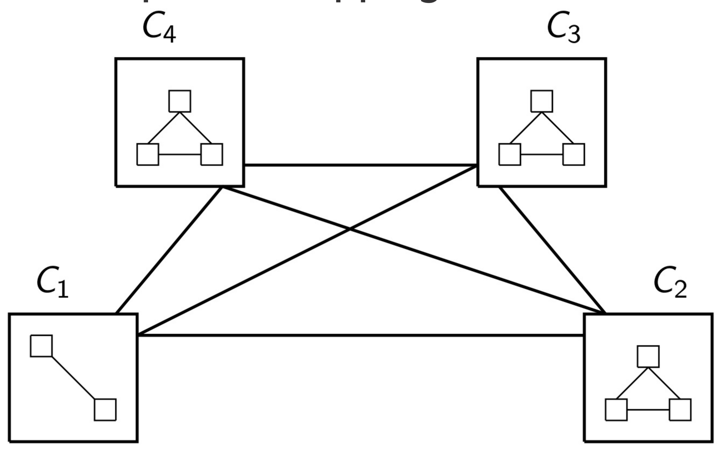
Combinatorial Testing Methods for Composed Systems

Dimitris E. Simos, Ludwig Kampel, Bernhard, Garn, Murat Ozcan

Combinatorial Methods for Modelling Composed Systems

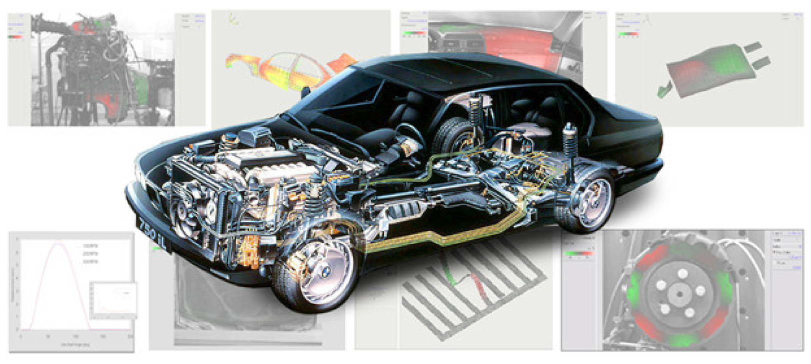
Complex systems composed of multiple components
Emerging in vast numbers in modern information society:

- Contemporary software design relies on **modular software architecture**, making it better understandable and maintainable
- A **modern vehicle** is a composed complex system in itself
- **Communicating autonomous vehicles** are even more complex
- **Smart buildings** like hospitals, shopping malls, etc.



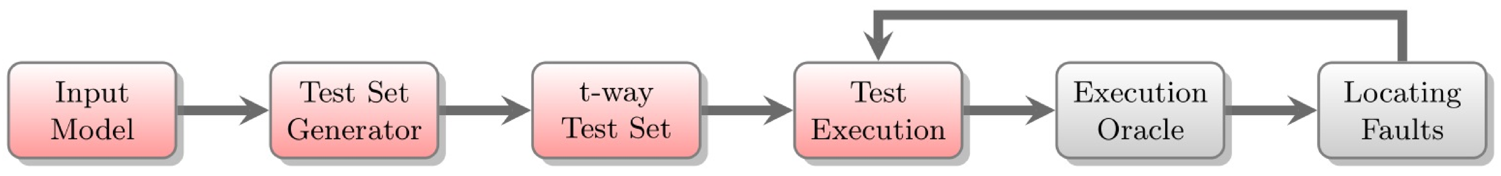
- Composed SUT with components C_1, C_2, C_3 and C_4, e.g., a car:

C_1: Wheels modelled via `producer`, `type`
C_2: Engine modelled via `fuel`, `drive-mode`, `filter`
C_3: Infotainment modelled via `streaming`, `audio`, `remote`
C_4: Communication modelled via `ip-version`, `connection`, `speed`

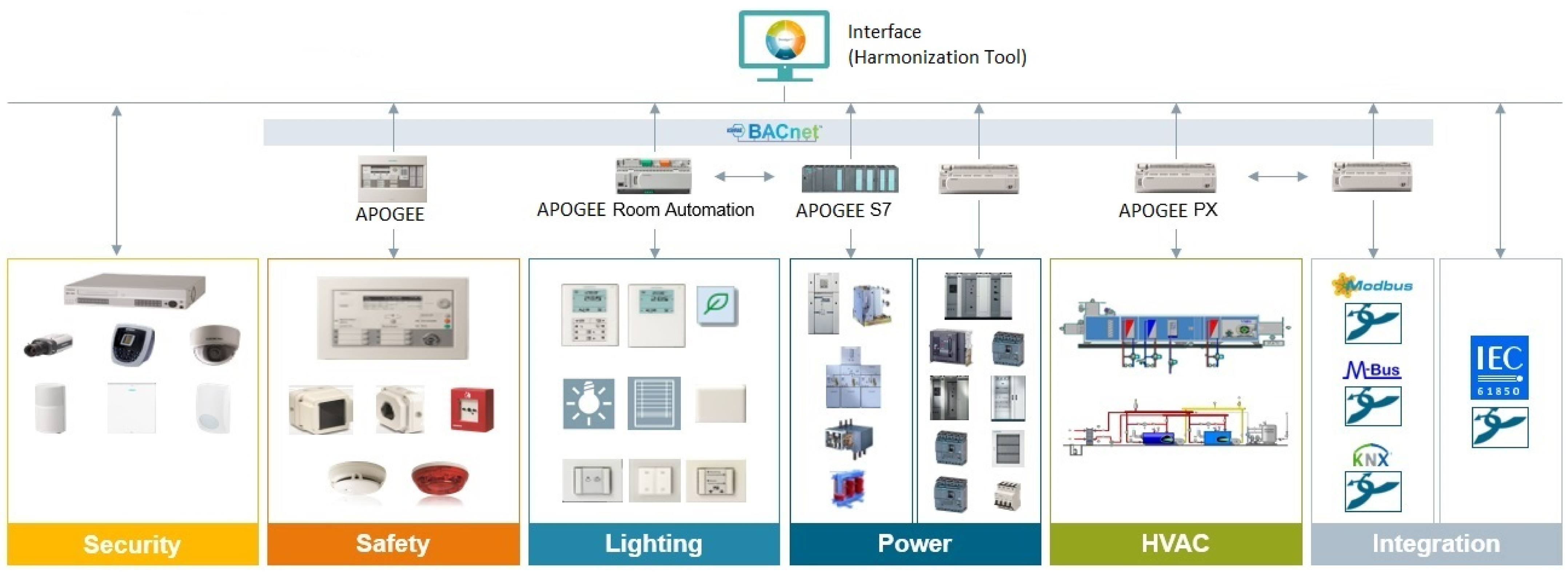


Methodology
Different in nature and in their application domains, these systems under test (SUTs) are **systems composed of sub-systems**:

- Devise a combinatorial model and a test suite for each sub-system
- Devise a combinatorial model for the unifying meta-system
- Apply a combinatorial construction to merge test suites of the components to a test suite for the whole SUT
- From theory we know: The **coverage of all t-way interactions is inherited** to the overall test suite



BACnet (Building Automation and Control networking) Protocol



- BACnet enables devices access via the network.
- Interoperability among different vendors' equipment.
- One operator interface to handle any device in the network.
- US, EU and ISO standard.

Combinatorial Methods for Testing BACnet

- BACnet models devices using an object oriented structure.
- Event Enrollment Objects (EEOs) provide an interface to communicate with devices.
- 5 million ways to configure an EEO.

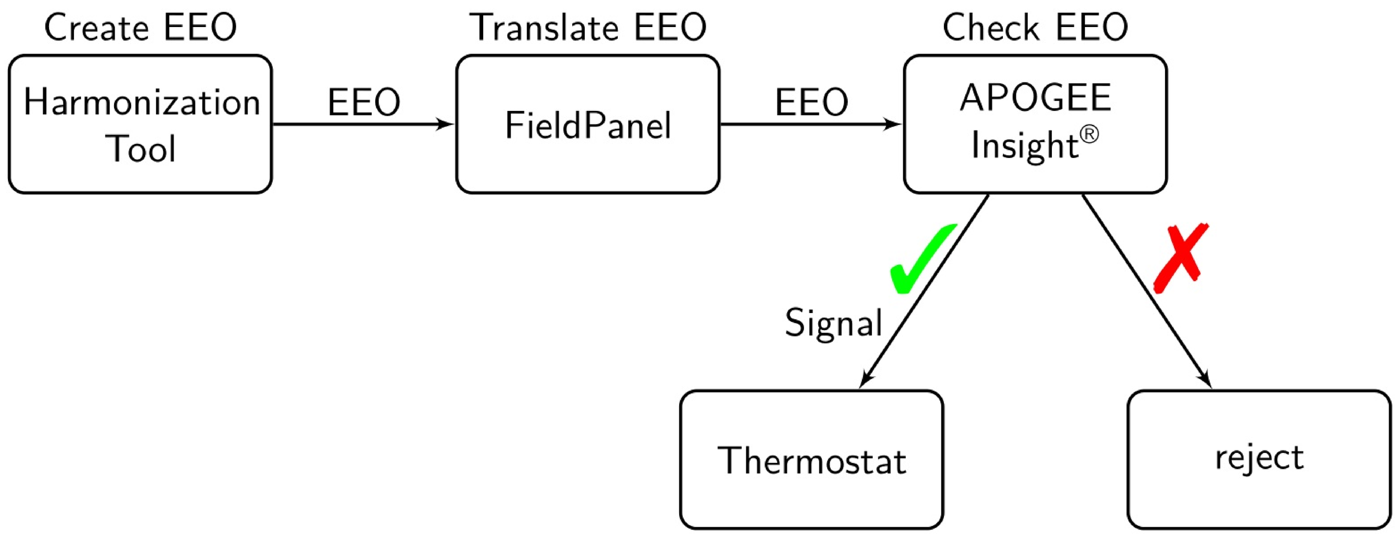


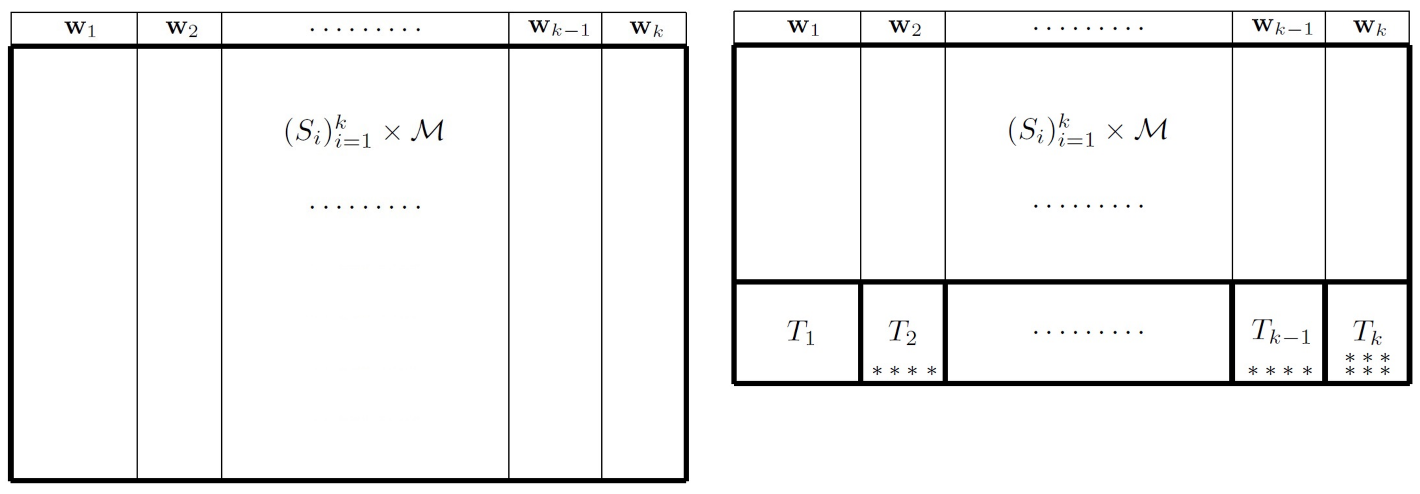
Figure 1: Example of the testing work-flow with APOGEE Insight® as BACnet client.

- (Nested) IPM for EEOs.
- Optimized test suite, maching the constraints of the application.
- Execute tests.

A Plug-in Construction for CAs Reflecting Composed Systems

- **Goal:** Construct CAs with more factors from CAs with less factors.
- **Idea:** Adapt plug-in construction from classic design theory for CAs.
- **Methodology:** Make use of coverage inheritance.
- **Application:** Combinatorial Testing for composed (Software) Systems.

Theorem. *Given an MCA* $\mathcal{M} = \mathsf{MCA}(N; t, k, (u_1, \ldots, u_k))$ *and two families* $T_i = \mathsf{MCA}(v_i; t_i, g_i, \mathbf{w}_i = (w_{i,1}, \ldots w_{i,g_i}))$ *and* $S_i = \mathsf{MCA}(u_i; t_i - 1, g_i, \mathbf{w}_i = (w_{i,1}, \ldots w_{i,g_i}))$ *of MCAs, for* $i = 1, \ldots, k$*. Then a* $\mathsf{MCA}(M; \tau, \sum_i g_i, (\mathbf{w}_1, \ldots, \mathbf{w}_k))$ *can be constructed, where* $M = N + \max_{i \in \{1,\ldots,k\}}\{v_i\}$ *and* $\tau = \min_{i \in \{1,\ldots,k\}}\{t_i, t\}$.



L. Kampel, B. Garn, and D. E. Simos. Combinatorial methods for modelling composed software systems. In *2017 IEEE International Conference on Software Testing, Verification and Validation Workshops (ICSTW)*, pages 229–238, 2017.
Dimitris E. Simos, Ludwig Kampel, and Murat Ozcan. Combinatorial methods for testing communication protocols in smart cities. In Roberto Battiti, Mauro Brunato, Ilias Kotsireas, and Panos M. Pardalos, editors, *Learning and Intelligent Optimization*, pages 437–440, Cham, 2019. Springer International Publishing.

Bundesministerium Klimaschutz, Umwelt, Energie, Mobilität, Innovation und Technologie
Bundesministerium Digitalisierung und Wirtschaftsstandort
FFG Forschung wirkt.
wirtschafts agentur wien Ein Fonds der Stadt Wien
FWF Der Wissenschaftsfonds.
netidee OPEN INNOVATIONS
SBA Research (SBA-K1) is a COMET Centre within the framework of COMET – Competence Centers for Excellent Technologies Programme and funded by BMK, BMDW, and the federal state of Vienna. The COMET Programme is managed by FFG.