

On the Evolvable and Traceable Design of (Under)graduate Education Programs

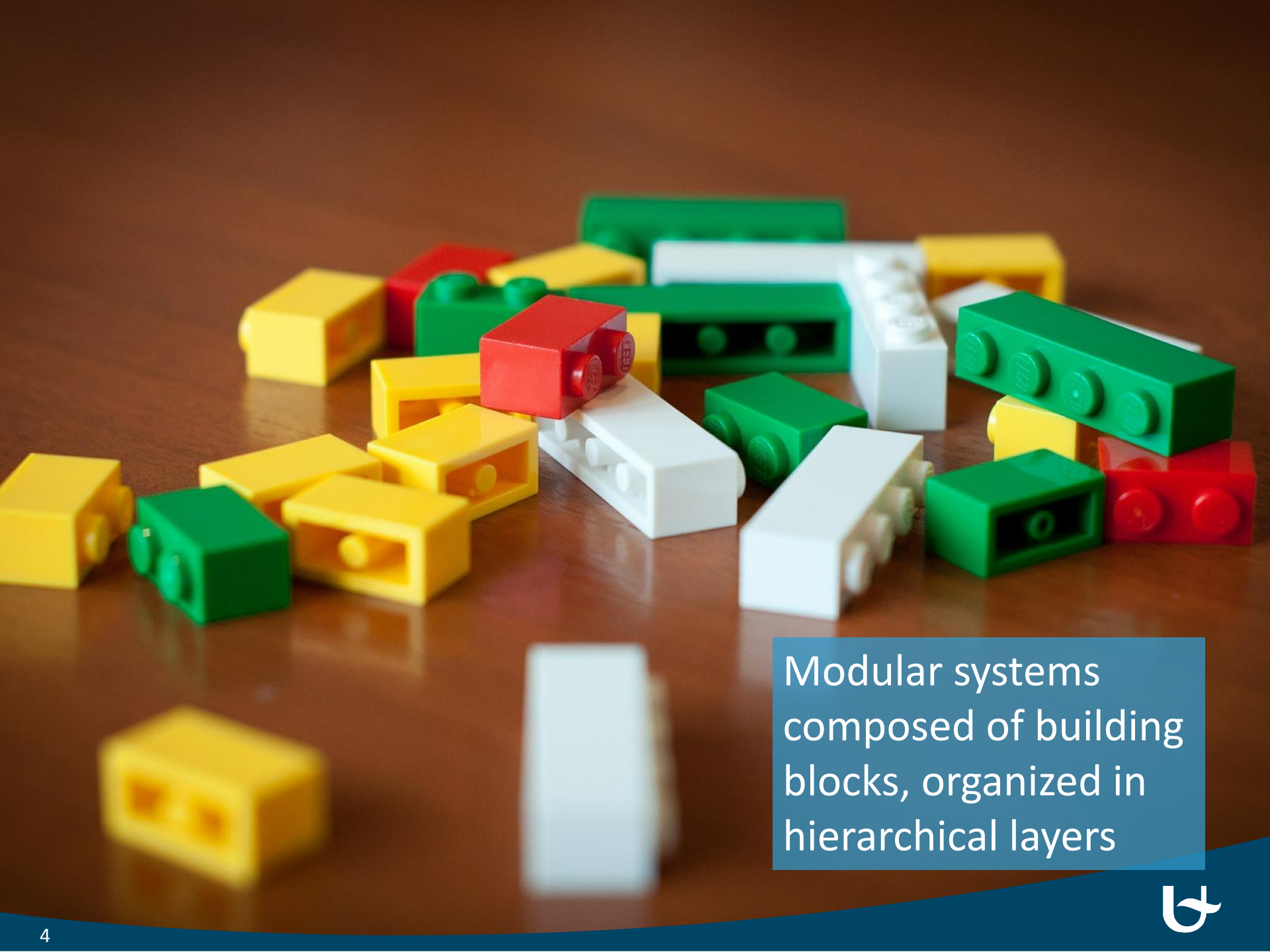
Gilles Oorts, Herwig Mannaert, Peter De Bruyn, Ilke Franquet

Part 1

Studying Modular System Design

Studying the Design of Modular Systems

- Studied modular systems:
 - Software
 - Enterprise Architecture
 - Business processes
 - Accountancy
 - **Study programs**
 - ...
- Based on traditional **engineering** and **Normalized Systems principles**
 - This paper: *modularity, cohesion, coupling, evolvability* and *traceability*



Modular systems
composed of building
blocks, organized in
hierarchical layers

Part 2

Study Program Design

Study Programs

- “The prescribed composition and succession of courses for a student to take in order to graduate”
- Ever growing **variety** and **flexibility**
 - Minors/majors
 - Optional courses
 - Flexible study programs (e.g., 70% eligible courses)
- Large amount of **dependencies**
 - Constraints of prerequisite courses
 - Courses taught in several study programs
- This complicates **design** and **management** and hampers **change**

Study Program Design principles

1. Modularity

- Only 1 level of hierarchy generally accepted: courses
- We propose to add:
 1. Learning-teaching tracks and sub-tracks
 - “Collection of courses within a study domain and sub-study domain”
 - E.g., Business Economics (track) and Accountancy (sub-track)
 2. Course parts
 - New lowest level of modularity
 - Makes them **accessible** and **evolve** freely
 - Can be taught/examined and used to compose new courses for specific study programs/students

Study Program Design principles

2. Coupling

- Study programs are prone to high amounts of coupling
- Dependencies:
 - Prerequisite courses
 - Courses taught in multiple study programs

Study Program Design principles

3. Evolvability, limited by:

- **Ripple/Combinatorial effects**

- Changes in the composition, sequence or prerequisite requirement of courses

- **Transition effects**

- Financially and practically to support both the old and new version of a study program
- Phasing out a study program no longer possible
 - Increasingly flexible study programs
 - Interrelated study programs
- Measures for students due to flexibility complexity

Study Program Design principles

Numerical example, transition effect:

	No added structure	With learning-teaching tracks
Study program structure	30 courses	6 tracks with 5 courses each
States of study progress	$2^{30} \approx 1$ billion situations	$6 * 2^5 = 192$ situations

-> adding hierarchical structure to study program design reduces variation complexity

Study Program Design principles

4. Traceability & cross-cutting concerns

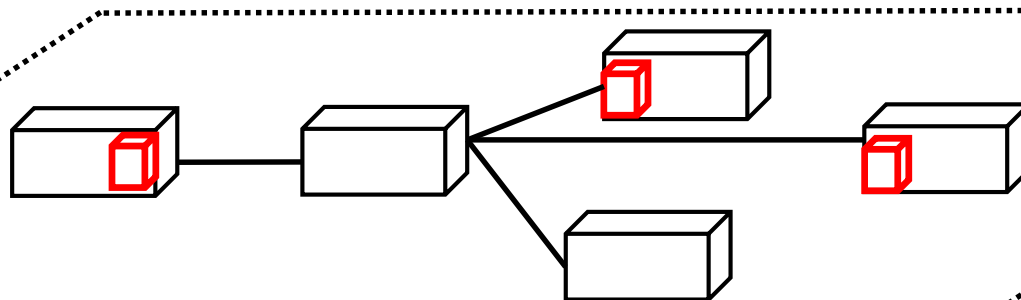
- Quality and environmental requirements
 - **Specified** at general study program level...
 - ...but **realized** and **monitored** at course level

= problem of **traceability**

- **Cross-cutting concerns** (Normalized Systems theory)
 - “Often occurring functionality that manifests itself in a different dimension”
 - Quality/environmental requirements = cross-cutting concerns
 - E-learning, case studies, assignments, internationalization,..

Study Program Design principles

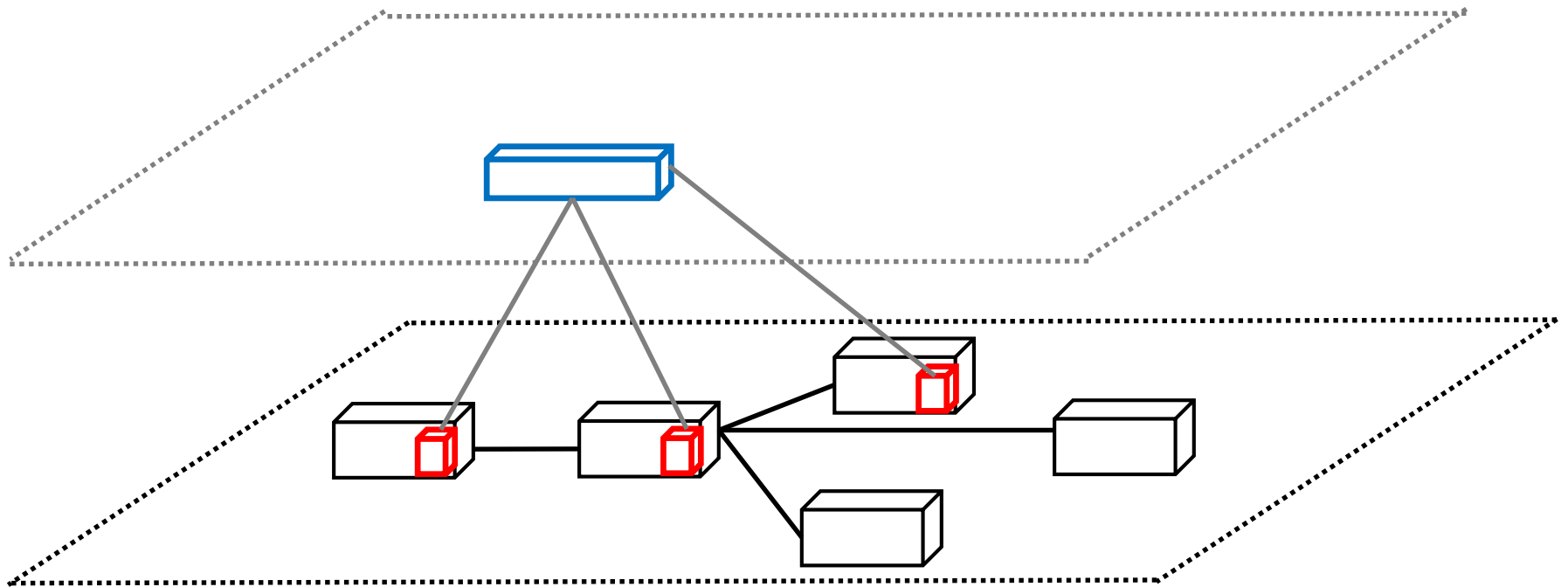
- *Implementing* cross-cutting concerns
- *Three* implementation solutions:
 1. Decentralized



Disadvantages: - Duplication functionality/No economies of scale
- Changes need to be applied individually

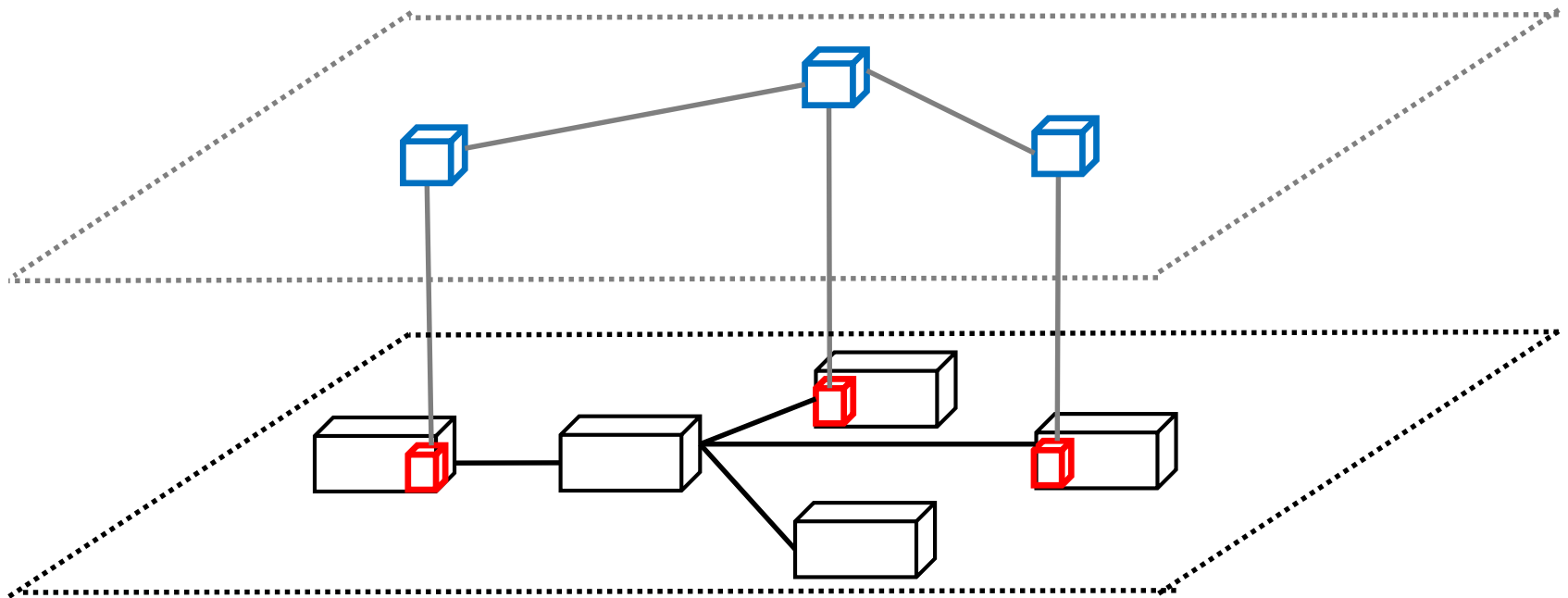
Study Program Design principles

- *Implementing* cross-cutting concerns
- *Three* implementation solutions:
 1. Centralized provider
 2. Centralized provider



Study Program Design principles

- *Implementing* cross-cutting concerns
- *Three* implementation solutions:
 3. Coordinated working groups



Part 3

Case studies

Case 1: University of Antwerp

- Faculty of Applied Economics
 - 3,250 students
 - Five Bachelor study programs, two Master programs
 - 258 courses
- **Re-design** of study programs two years ago
- **Nine learning-teaching tracks** and **27 sub-tracks**

Case 1: University of Antwerp

Learning/Teaching track	Sub-track
General economics	Fundamentals
	Policy
	Accountancy
Business economics	European and international business
	Finance
	Marketing
	Strategy and organization
	Transport and logistics
Engineering	Fundamentals
	Sustainable technology
	Supply chains and operations
Information systems	Fundamentals
	Engineering and architecture
	Governance and audit

Case 1: University of Antwerp

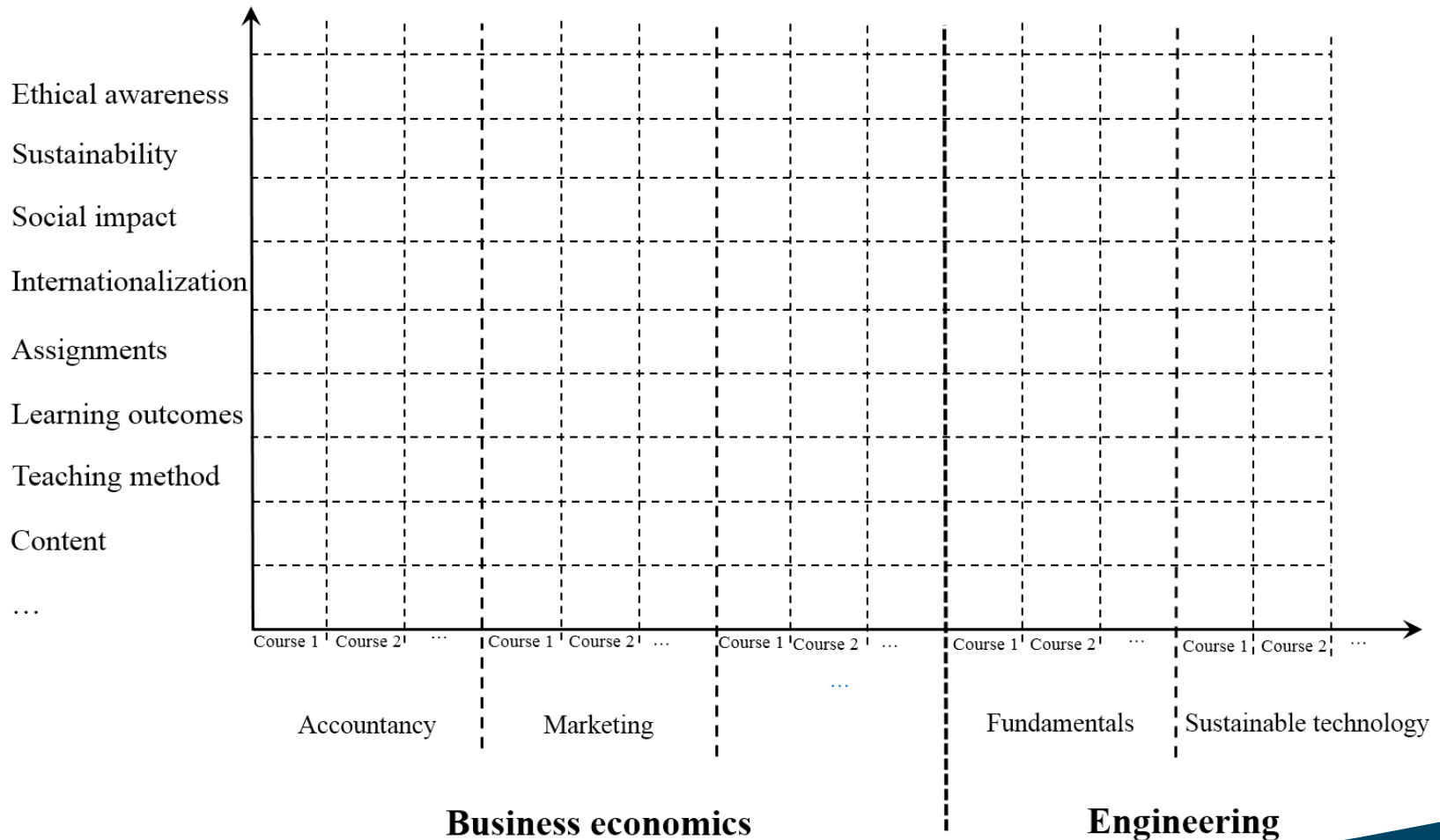
Linking courses in a (sub)track to study programs

Course code	Course name	Credits	AE-BE	AE-EP	BE	MIS
1101TEWAEC	Introduction to general economics	6	B1	B1	B1	B1
1103TEWVSG	European and international law	6	B1	B1		
1201TEWAEC	Micro economics	6	B2	B2	B2	B2
1201TEWKOO	History of economic thought	6	B3	B3	B3	B3
1202TEWAEC	Macro economics	6	B2	B2	B2	B2
1301TEWECEB	Contemporary economic and political history	6		B3		

Sub-track 'General economics - Fundamentals'

Case 1: University of Antwerp

Tracing study program level cross-cutting concerns to courses in a (sub)track



Case 2: NGO study programs

- NGO trains people in search for employment
 - Different setting, still educational institution
- Studied how the measures of case 1 could be applied
- Findings:
 - Learning-teaching tracks can be defined
 - Dependencies between courses are also present
 - Cross-cutting concerns were identified
 - Similar (e.g., ethical awareness, end competence monitoring,...)
 - And different (e.g., showing contribution to job compliant attitudes)

Conclusion

1. Managing/changing study programs is more **complex** than one might think
2. Study programs exhibit large amount of **dependencies**
3. This leads to both **ripple effects** and **transition effects**
 - Hampering or preventing changes
4. However, can be designed as a modular system with specific **hierarchical structures** to reduce impact
 - Cross-cutting concerns, learning-teaching tracks, course parts

