VICTORIA UNIVERSITY OF WELLINGTON
TE HERENGA WAKA

UNIVERSITÄT DUISBURG ESSEN

Offen im Denken

FMML^{x} and DLM

A Contribution to the MULTI Collaborative Comparison Challenge

Thomas Kühne, **Pierre Maier**

Overview of Presentation

1. Introduction and Challenge Description
2. FMMLx and DLM: Comparative Analysis
    1. Notation and Tool Support
    2. Relationships between Associations
    3. Level Concept
    4. Deep Characterization
    5. Separation of Modeling Concerns
3. Conclusions

MULTI Comparative Comparison Challenge

- huge diversity of MLM approaches, desire for more unification
- CC Challenge: Mobile Phone Factories
- both language already participated in the challenge
  - DLM and LML in 2022; FMML^{x} and LML in 2023
- list of 13 requirements provided as part of the description

"A company owns factories" (Req. 1b)

"A mobile phone device has an IMEI" (Req. 7b)

"S400_2 conforms to the S400 model" (Req. 13b)

"S400 is a mobile phone model" (Req. 11a)

1 Notation and Tool Support

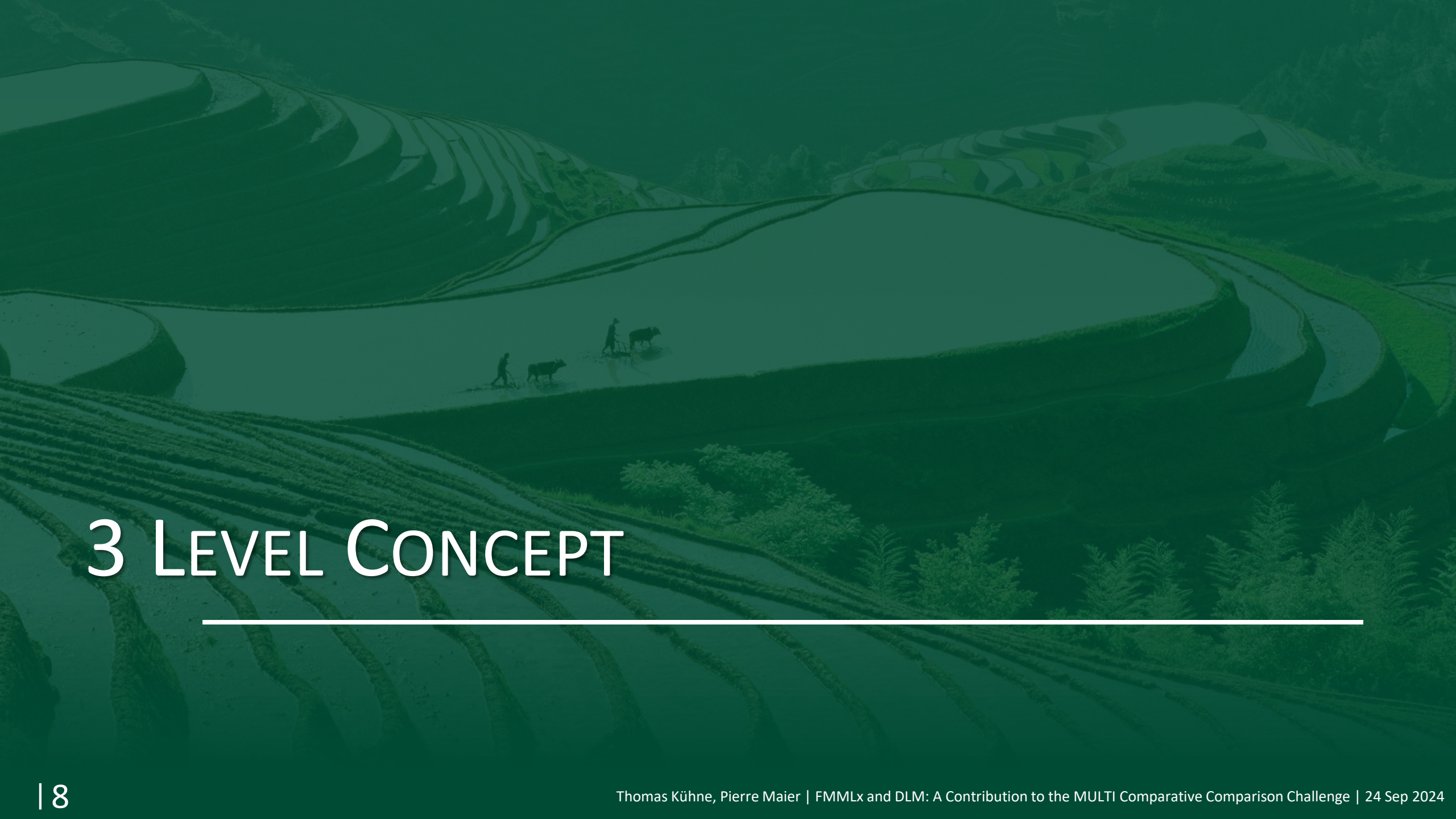
1 Notation and Tool Support

FMML^{x}	DLM
• UML-like notation with level-specific color encodings • possibility to define custom notation with *Concrete Syntax Wizard*	• UML notation with colored classification dimensions
• fully supported by XModeler^{ML} • models can be instantiated and executed at run time	• partial implementation in ConceptBase

2 Relationships between Associations

2 Relationships between Associations

Requirements	FMML^x	DLM
3) A factory … a. produces devices b. supports a list of device models c. can only produce devices that conform to (are of) supported device models	• Req. 3a and 3b realized with *association types* (allows for custom notation) • Req. 3c addressed by *association dependency*	• Req. 3a and 3b realized with associations • Req. 3c addressed by *association specialization*



3 Level Concept

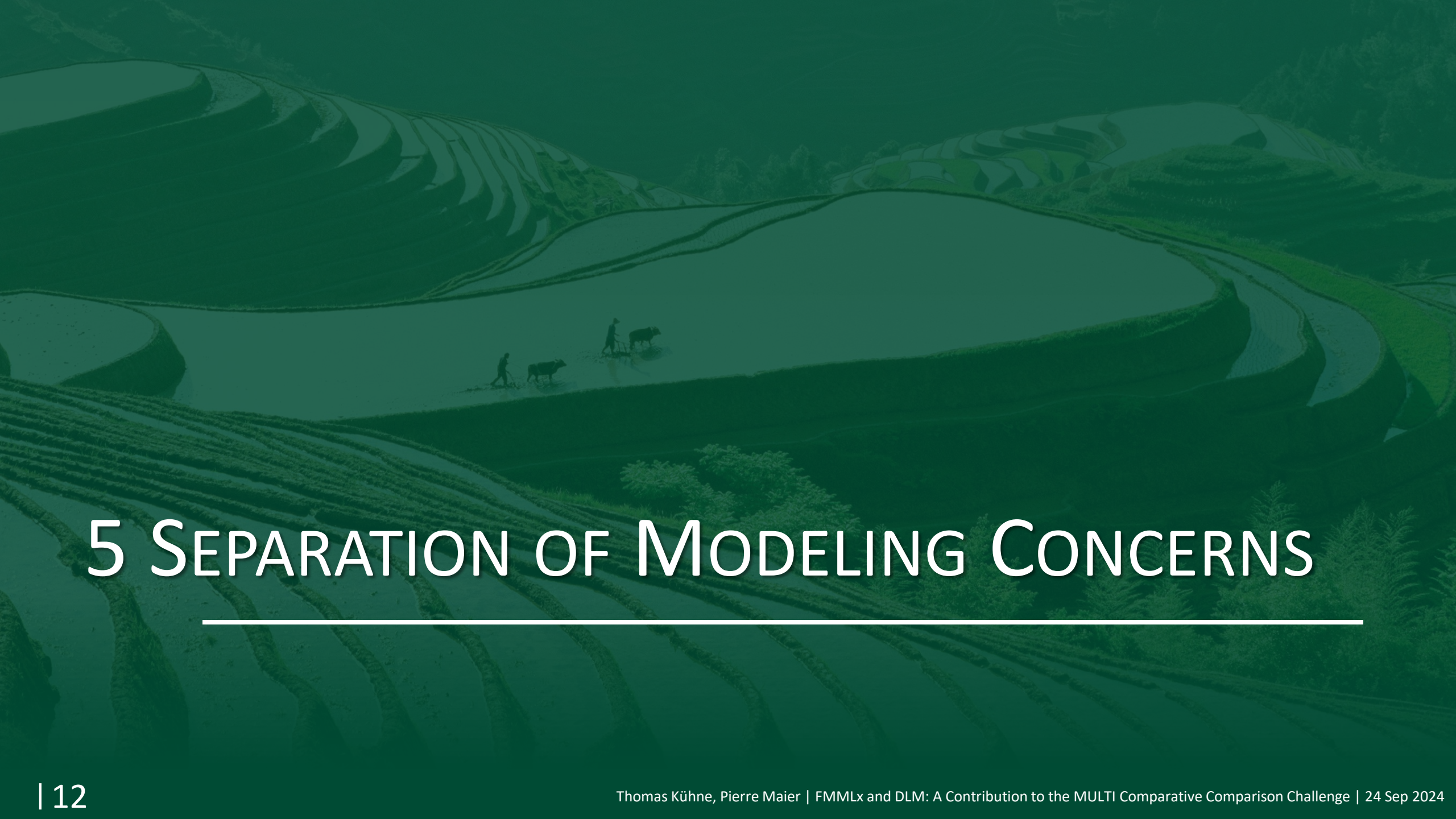
3 Level Concept

FMML^x	DLM
• pure generalization is intra-level relationship	• pure generalization is intra-level relationship
• cross-level associations supported	• cross-level associations supported
• inter-level relationship: concretization *OR* instantiation (between L0 and L1)	• inter-level relationship: instantiation
• concretization relationships might lead to counterintuitive model interpretations BUT requires fewer elements	• strict separation between instance-of and specialization relationships counteracts counterintuitive model interpretations BUT might require more elements

4 Deep Characterization

4 Deep Characterization

FMML^{x}	DLM
• *deferred instantiation*	• *deep instantiation*
• absolute target-level specification	• relative target-level specification



5 Separation of Modeling Concerns

5 Separation of Modeling Concerns

FMML^x	DLM
• multiple views for one diagram (separated by layout) • multiple diagrams for one model	• classification dimensions

Conclusions

COMMONALITIES

- overlap in core concepts (class, generalization, etc.) and notation
- similar issues addressed differently
  - association dependencies and association specialization
  - views & diagrams vs. classification dimensions
  - approaches may complement each other

DIFFERENCES

- sharp difference of inter-level semantics
  - approaches have different priorities
- different target-level specification principle
- FMML^{x} in some aspects more advanced
  - complete tool support, is executable, offers custom notation

TAKE AWAYS

- approaches can learn from each other
- difference in modeling language sometimes result of different priorities or stylistic choices
- open question: what are trade-offs of the target-level specification styles?