New Summary for Final Exam here:

https://docs.google.com/document/d/1KmLTGZ-qC97uGJxvETfvetCAnbMzcMUgVDGyPVdN80A/edit

Please feel free to edit & comment

Paper 1 - **Management control systems as a package - Opportunities, challenges and research directions**

There are several reasons why studying the MCS package patron is important: Firstly, MCS do not operate in isolation, if the connections between various MCS are not coordinated, then each of the considered MCS components relate to studied contingent variables will out bring erroneous conclusions. Secondly, studying these systems individually may influence any conclusions we can draw, if the use and impact of a new MCS element is related to the functioning of the existing broader MCS package.Thirdly, a major focus of MCS theory is how to design MCS in order to produce the desired outcomes.

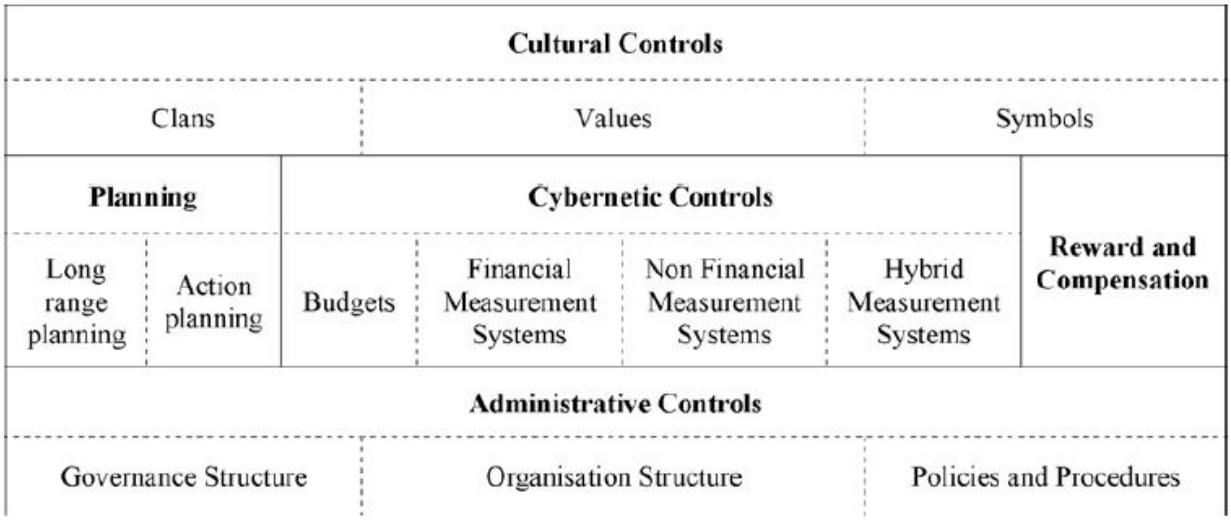


Fig. 1. Management control systems package.

Example University:

Clans: Fachschaft, Professoren, Studiengruppe
Values: Wissen vermitteln, Welt verbessern, Nachhaltigkeit
Symbole: [!] , Hausordnung

Long range planning: Tiss, Studienplan
Action planning: Tuwel

Budgets: SAP
Financial Measurement Systems: SAP

Non Financial Measurement Systems: Umfragen
Hybrid: ??

Reward and Compensation: Zeugnisse, Stipendien

Governance Structure: Board of Deans, Dekanat
Organisation Structure: Organigramm
Policies: Studienplan, Datenschutzrichtlinie

Main Idea:

- Decision making system

Difference between

- **Decision-Making**: Provide information to support decision making
- **Control**: Influence activities and behaviors of employees

Five types of controls:

Cultural Controls:
These are placed above, because they give a **frame** and are **hard to change**. Comparing to administrative controls, cultural controls are **broad** yet offer **subtle control**

- **Clans**: Support groups where skills are developed and work as a unit
- **Values**: Senior managers define values that should be adopted by employees
- **Symbols**: Visuals such as dress codes, workspace design, colors, …

Planning:
It sets out the goals of the functional areas of the organisation, thereby directing effort and behaviour. Further it provides the standards to be achieved in relation to the goals, and clarifies the level of effort and behaviour expected from organisation members.

- Long range planning: **strategic** focus with goals + actions for the medium and long run
- Action planning: **tactical** focus that focus on the **immediate** future

Cybernetic Controls:
Cybernetic controls enables to **quantify** underlying problems, activities or systems, they set **standards of performanc**e and **feedback** processes, and they are able to modify the system's behaviour or underlying activities. Presented in a temporal order from left to right

- Budgets: **planning** behaviour and **evaluating** performance against budget plansgg2sf
- Financial Measurement Systems:
- Non- Financial Measurement Systems:
- Hybrid Measurement Systems:

Reward and Compensation:
Motivation, Increase the performance of individuals. Intrinsic vs. Extrinsic
Link effort to task can influence in: effort **direction**, effort **duration**, effort **intensity.**
The main goal of this systems is to motivate and increase performance of members of the organization by achieving goal congruence. (Monetary) Incentives help influence behavior and this way managers can control the effort direction, duration and intensity of their employees.

Administrative Controls
direct behaviour, monitor, accountability

Organisation Design:
Contact and relationships, reducing variability of behaviour, Organigramm

Governance Structure:
Meetings, Agendas, Deadlines, Governance Process
Control behaviour of employees

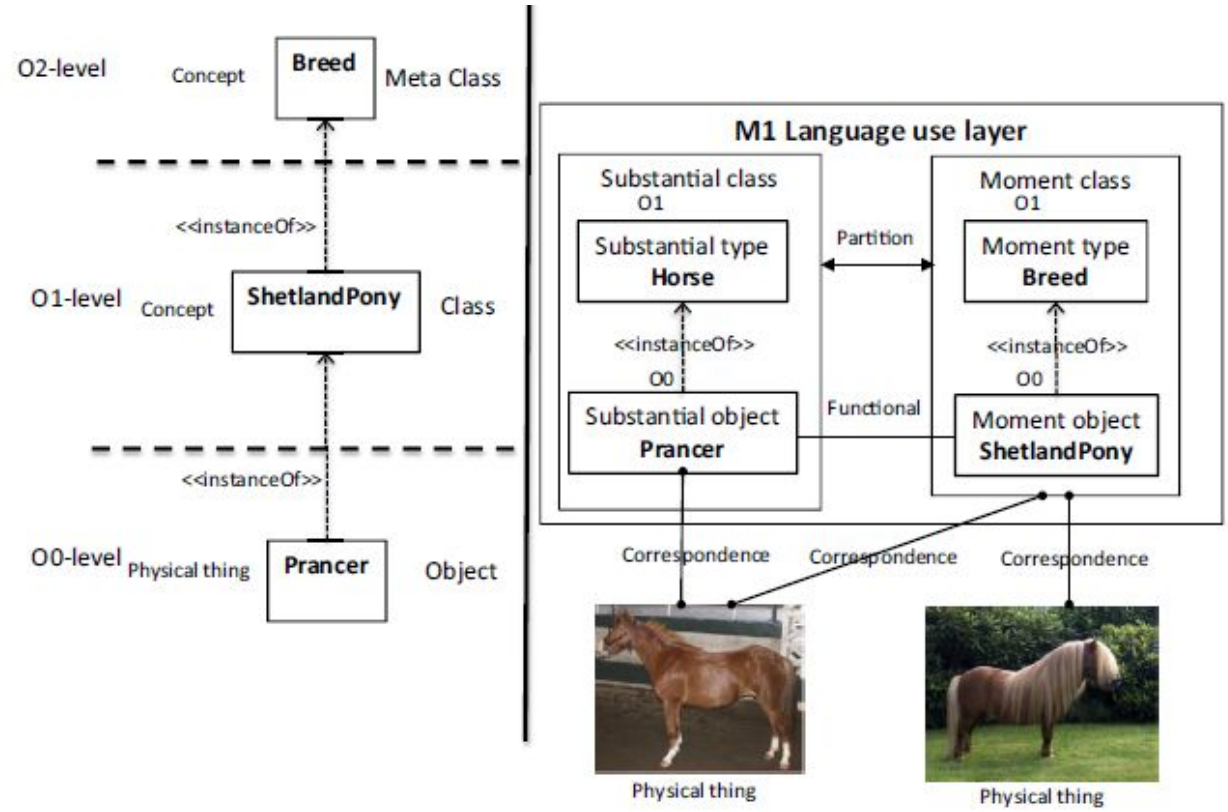
Policies and procedures:
Specify Processes and behaviour, Activity Diagrams
Open questions:

- How does this package work in a real organisation?
- Which component influences how?
- How is the order of components influencing?

A good control system is composed of management control + financial control

Why study as a package?
The MCSs are grouped in a package, because the different systems are, in practice, often introduced by different parties in the organization and at different times, so that they are not really a homogenous group (in which case the whole system could be called a MCS), but rather a package of systems.

Paper 2 - Please feel free to edit & comment



Bsp:

Moment Class
Moment Type: Study Program
Moment Object: VU (LVA Number, Course Name, ECTS, …)

Substantial Type: Course
Substantial Object: IT Based Management

Physical Thing: IT Based Management

M3 (Meta Object Facility) Metamodel:
M2 Definition of the language
M1 Usermodel
M0 Run-Time Instances

Moment class
universal that can be associated with
a Substantial Class, which means that a Moment Individual can be
associated to many Substantial Individuals.

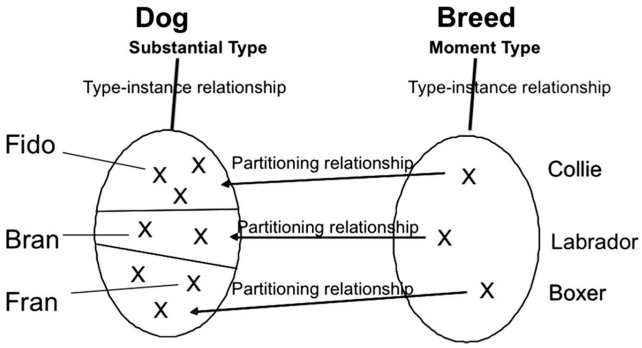
Substantial class
classes where the individuals have a 1:1 correspondence to substantial
things (physical things) and can reasonably be called ‘kind’.

Substantial Object
Represent real object

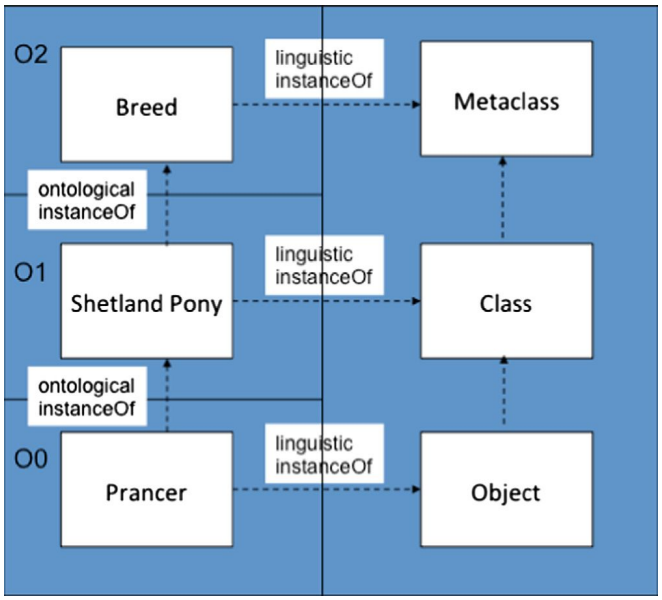
Moment Object
Represent Characteristics

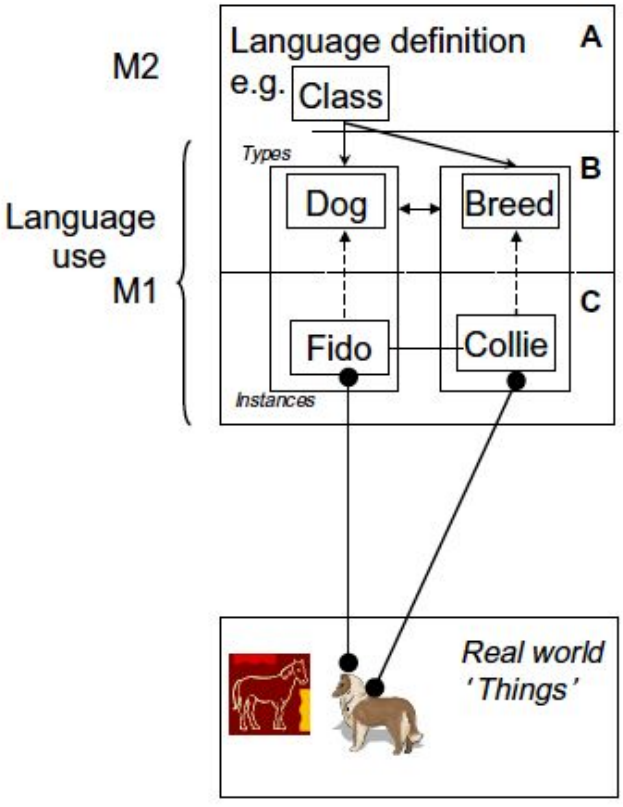
Functional Relationships
Functional relationship links Substantial Object to Moment Object, and the Substantial Object has the characteristics of the moment Object

Partition Relationship



Linguistic Meta-Modelling
Ontological Meta-Modelling
Orthogonal Classification Architecture





Key for Relationships

- Conformance Relationship between the definition (meta) level of a language and the language
- Partition Relationship between two classes
- Type-instance Relationship between the type and instance level of a class
- Correspondence Relationship between objects and things
- Functional Relationship between objects of different classes

Paper 3 - Please feel free to edit & comment

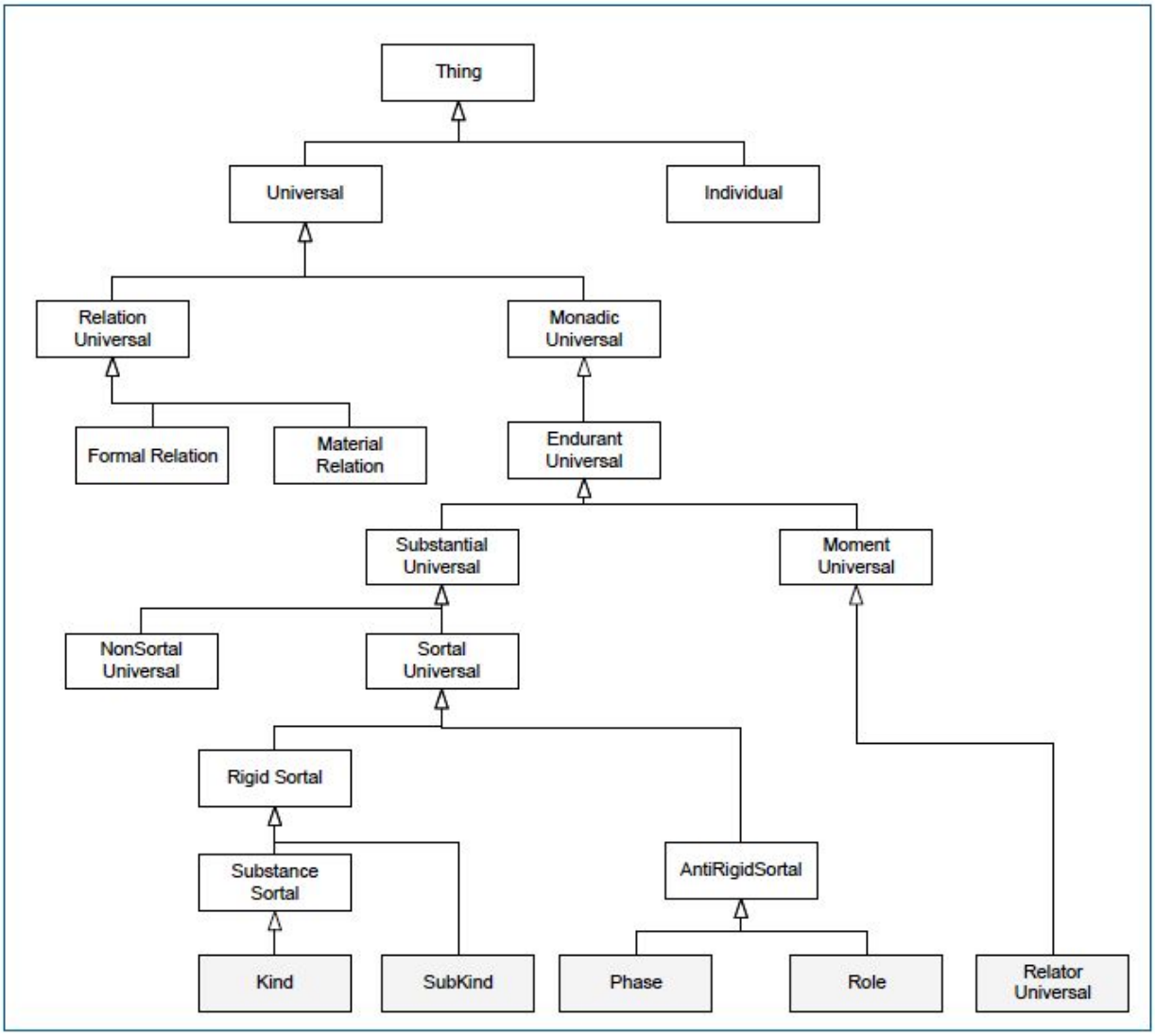
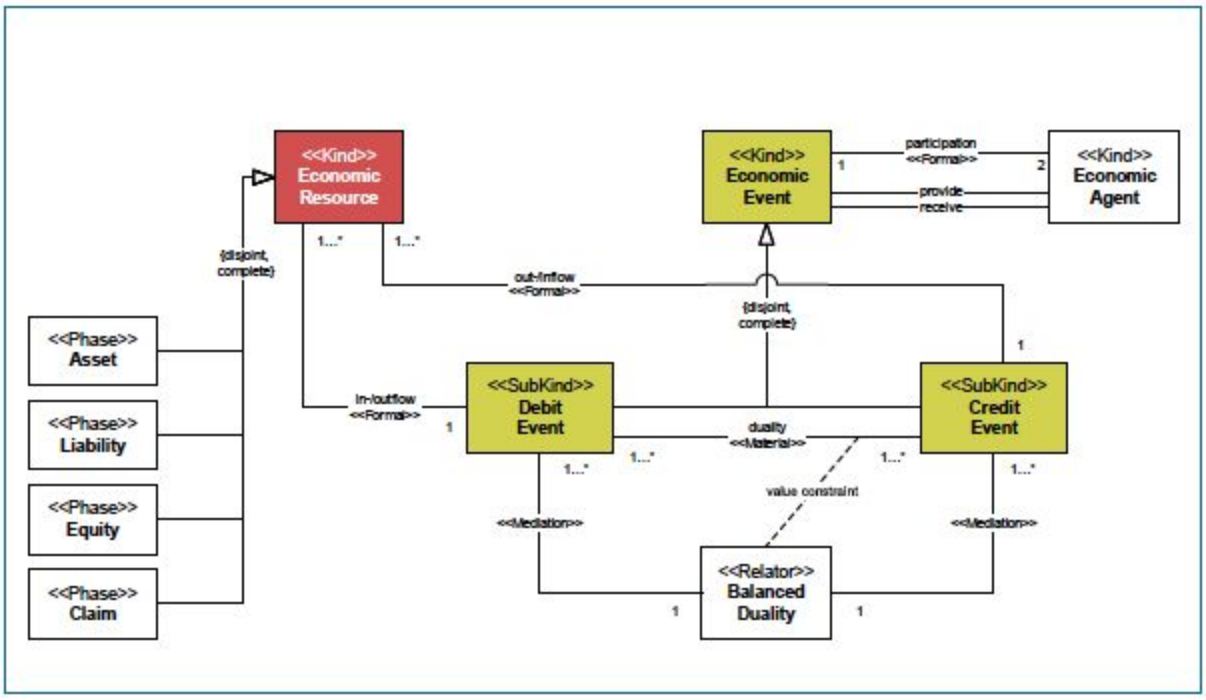


Figure 10. Relation types and (monadic) entity types – UFO excerpt

OntoUML language as an UML extension

Kind: Rigid and provide Identity
SubKind: Rigid but inherit Identity, Partitioning
Phase: Provides Identity and might exist by itself (in contrast to a Role), Anti-Rigid
Role: Has no identity and has to be linked to a Sortal, Anti-Rigid
Relator Universal: Mediated between two relationships



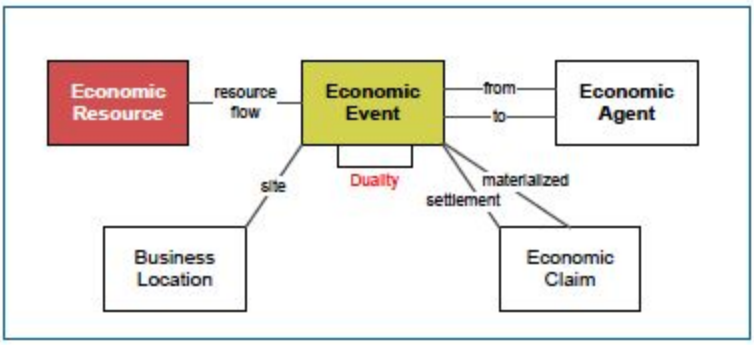
REA = Resources (R) are exchanged in economic events (E) between economic agents (A)
ALE= Asset Liability Equity
REA-based ALE accounting = REA has deficiencies with respect to ALE, therefore extension
OntoREA = REA-based ALE Accounting model in OntoUML
OntoUML is an UML extension that incorporates the ontological meta-properties and makes them accessible in UML class diagrams via stereotypes. entity-relationship (ER) language by covering ontologically differentiated types of entities and types of relationships

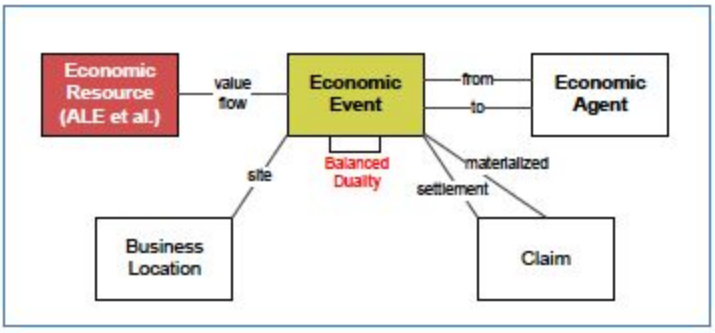
Four-Category Ontology by Lowe

1. UML
2. OntoUML = UML + ER + Ontology
3. REA Accounting = OntoUML + REA
4. OntoREA Accounting= REA Accounting + ALE

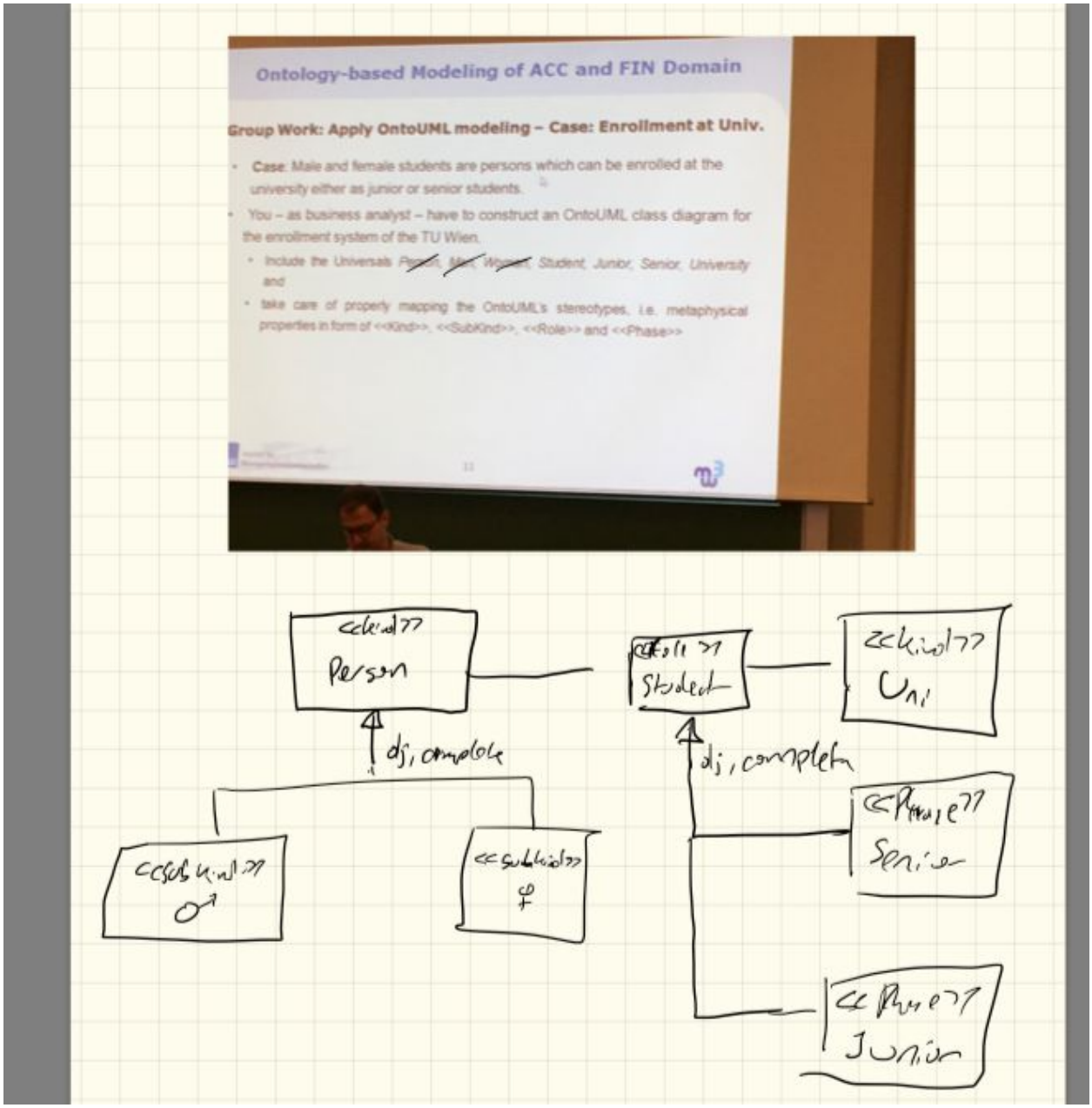
Duality:

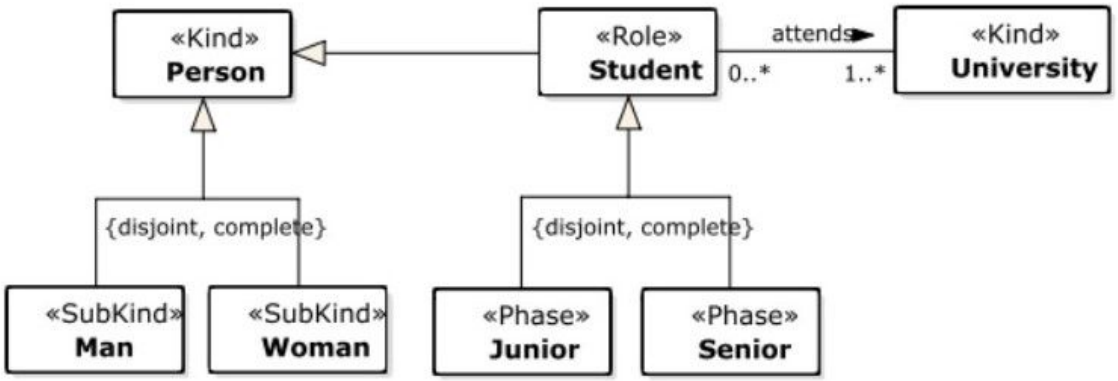
The duality relationship establishes **matches between increment and decrement events** but it does not **cover the value constraint** associated with the business transactions' debit and credit events, i.e. the balanced duality relationship


Economic Resource
resource flow
Economic Event
from
to
Economic Agent
site
Duality
materialized settlement
Business Location
Economic Claim


Economic Resource (ALE et al.)
value flow
Economic Event
from
to
Economic Agent
site
Balanced Duality
materialized settlement
Business Location
Claim

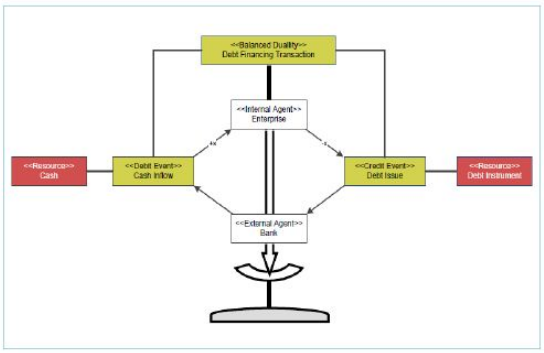
REA primitives	Mapping in Gailly et al. [11]	Mapping in OntoREA Accounting Model
Economic Resource	Role	Kind
• Asset/Liability/Equity/Claim	-	• Phase
Economic Event	Relator Universal	Kind
• Increment Event	• -	• SubKind
• Decrement Event	• -	• SubKind
Economic Agent	RoleMixin	Kind
• Inside Agent	• Role	-
• Outside Agent	• Role	-
Duality	Formal relation	-
Balanced Duality	-	Material relation

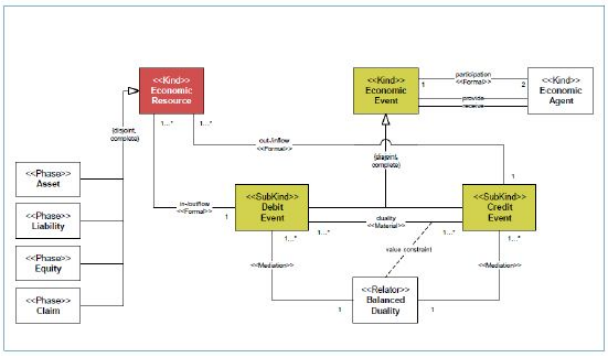

Ontology-based Modeling of ACC and FIN Domain
Group Work: Apply OntoUML modeling – Case: Enrollment at Univ.
Case: Male and female students are persons which can be enrolled at the university either as junior or senior students.
You – as business analyst – have to construct an OntoUML class diagram for the enrollment system of the TU Wien.
Include the Universals Person, Man, Woman, Student, Junior, Senior, University and
take care of properly mapping the OntoUML's stereotypes, i.e. metaphysical properties in form of <<Kind>>, <<SubKind>>, <<Role>> and <<Phase>>
Person
Student
Uni
dj, complete
dj, complete
Senior
Junior


«Kind»
Person
«Role»
Student
attends
0..*
1..*
«Kind»
University
{disjoint, complete}
{disjoint, complete}
«SubKind»
Man
«SubKind»
Woman
«Phase»
Junior
«Phase»
Senior

Group Work: Apply OntoUML modeling – Case: OntoREA ACC Model

- **Case**: „Ontology-driven“ modeling of the Resource/Event/Agent-“Ontology“
- Explain, what happens economically, if you take a loan from a bank?
- Describe the content of the Balanced Dualitity concept in a financing transaction and explain its modeling in the OntoREA accounting model.



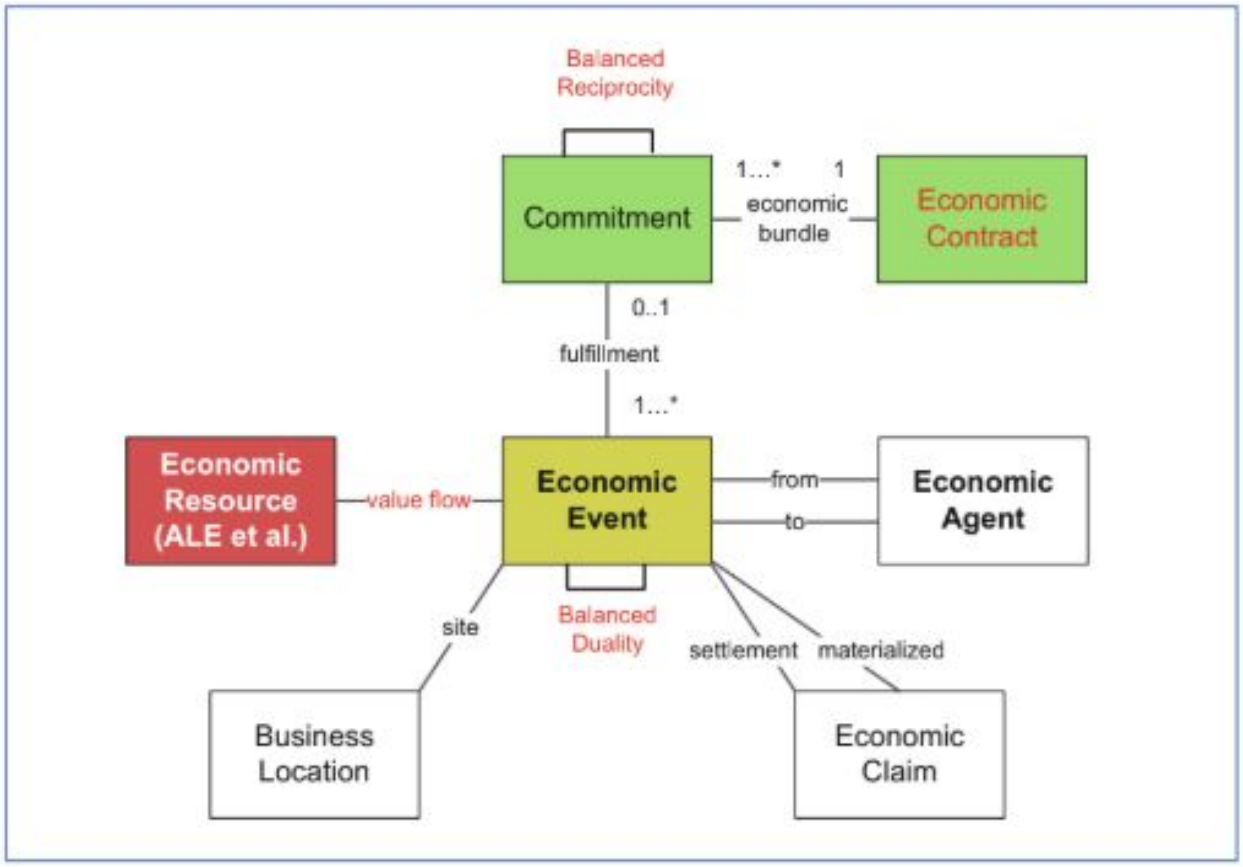


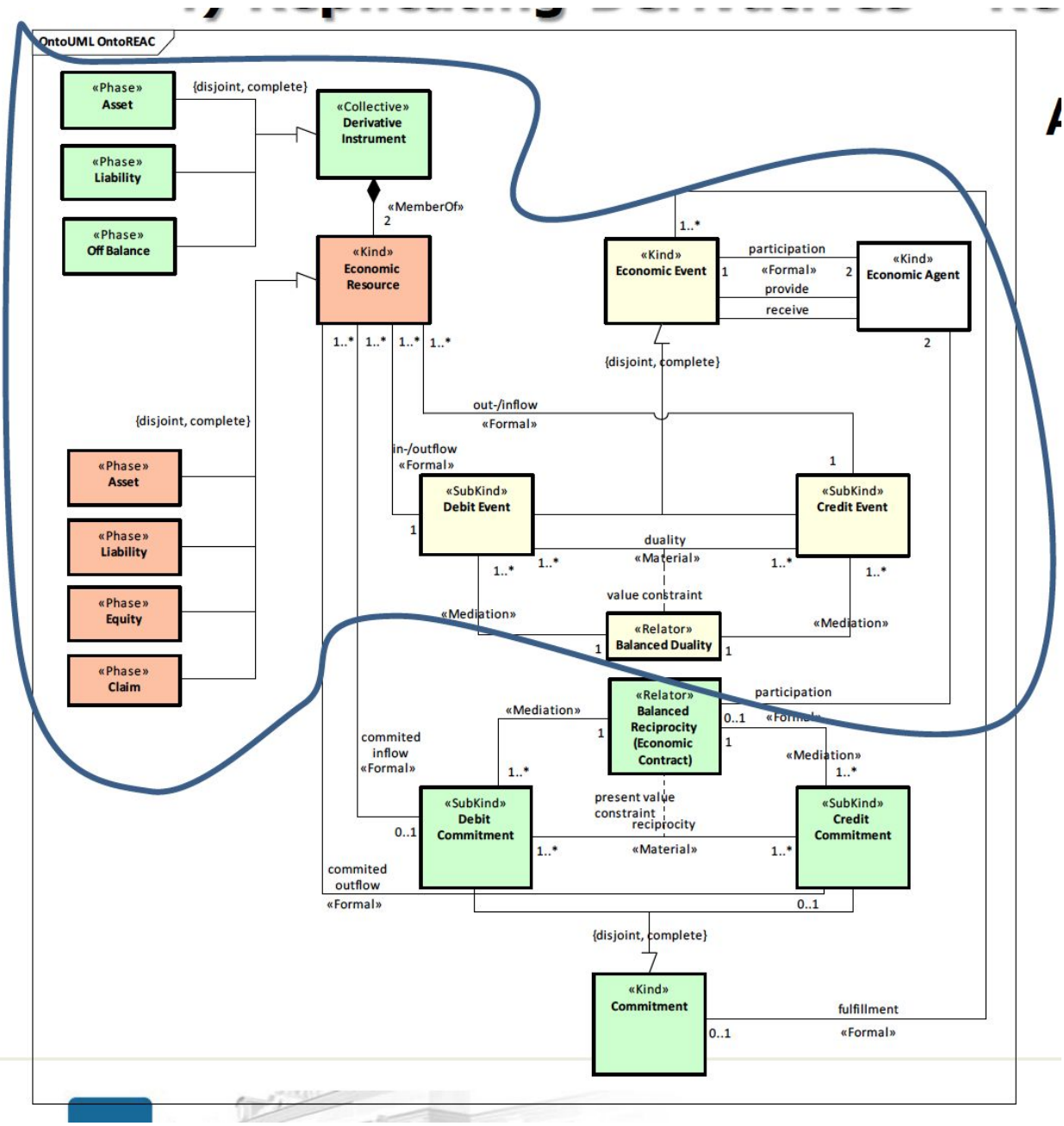
Bank Debits the Cash
I Credit the Cash

<<SubKind>> Credit Event = <<Phase>> Claim
<<SubKind>> Debit Event = <<Phase>> Liability

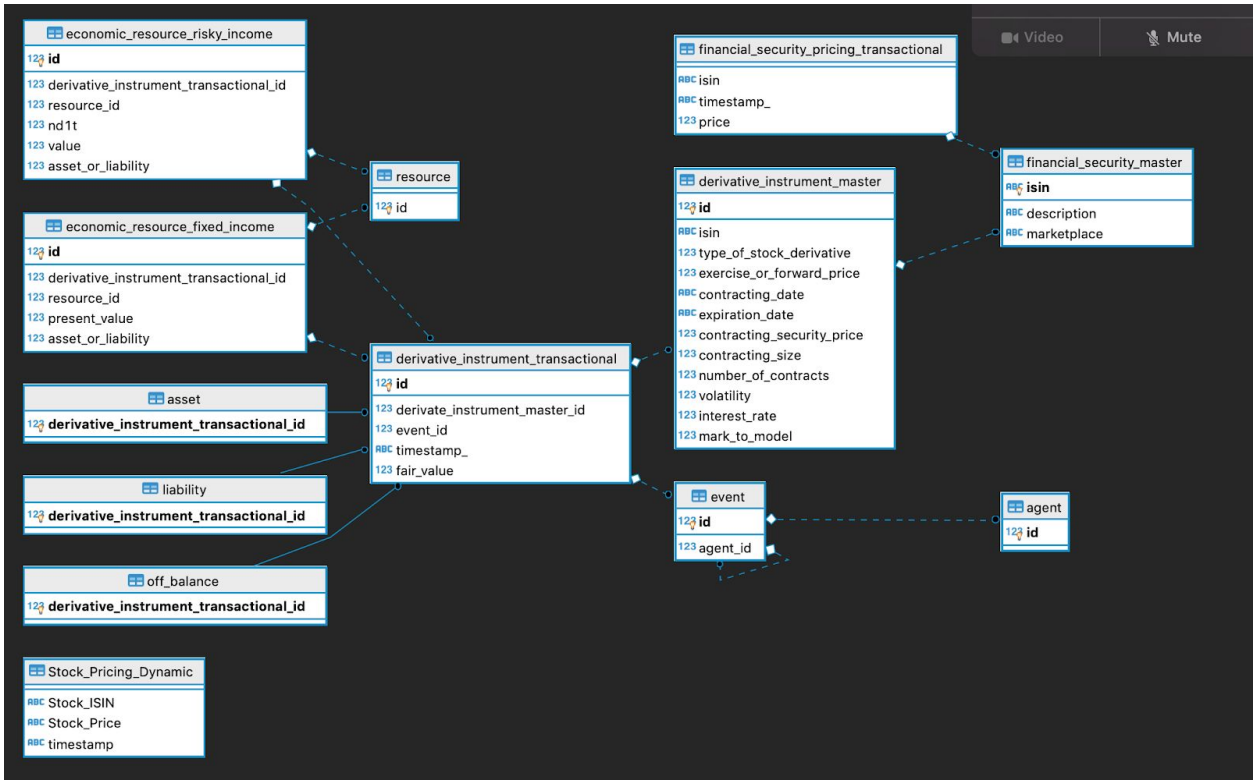
Debit Event and Credit Event are mediated by the Balanced Duality <<Relator>>
In contrast to the duality relationship, it also covers the value constraint associated with the business transactions’ debit and credit events, which is basically based on the principle of the conventional accounting equation.

Paper 4 - Please feel free to edit & comment





On top of the **Commitment** Class the relationship “balanced reciprocity” was added to express the future in- and outflowing event that underlie the debited and credited commitments and include a value constraint. Especially this relationship is used with the economic contract as truthmaker. Therefore the **<<Collective>> class** is integrated which has the same ontological meta-properties as the Kind universal but the Kind specifies individual universals whereas the **Collective specifies collections (e.g. Beatle**s). The distinction between **Kind and Collective results in a <<MemberOf>> parthood-relationship** because a group consists of group-members and these can leave or join the group without changing the group identity (no change of the Collective class).



Master Data vs. Transactional Data

Master data: one entry per derivative instrument, does not change over time

stores all the information that is linked to the good/service (isin, price …) and information about the contract framework (contract date, interest)

- id (identifying the data)

Transactional data: entries whenever a derivative instrument is changed by events stores all the data that is linked to the transaction that has to follow according to the signed contract.

- id (identifying the transaction)
- reference to the master data id
- Asset, Liability and Off_Balance inherit from Transactional Data, which means the Transaction data can be to a given timestamp an Asset, Liability or Off_balance.
- Additionally the Transaction Data is referenced by fixed income and risky income, which determine the type of income.
- Since the event and agent carry the transactions, these entities have in the data model a reference to Transactional Data.

Distinction of financial resources

Financial resources can be either an **asset or a liability,**

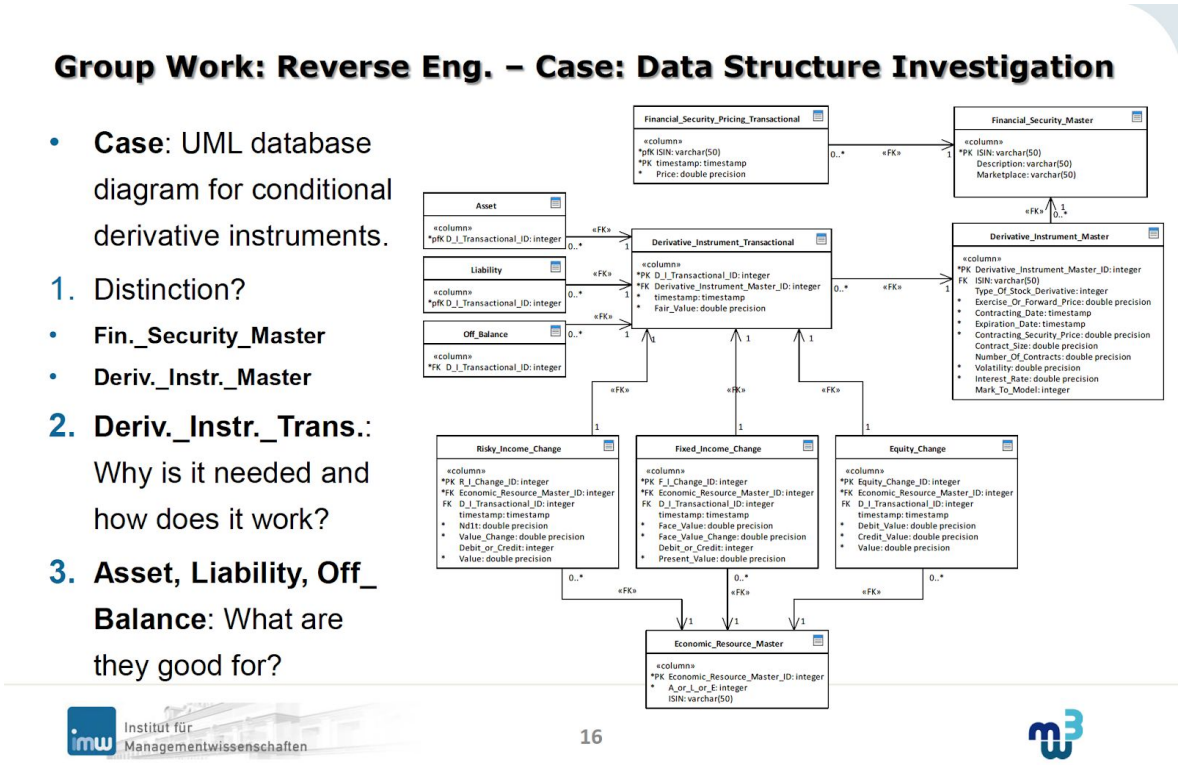
there is also a distinction in **risky and fixed income.** They inherit the id value of the resource table. The table derivative_instrument_transactional is in a MemberOf-relationship with the two tables from above. ...derivative instruments have two economic resource members, i.e. one asset and one liablity.” The id of the transaction can be stored in all three “Phase” tables asset, liability and off_balance, so the statement is fulfilled.

Necessity of asset/liability/off_balance

The Collective class Derivative Instrument is typified via a Phase partition that consists of the Phase classes Asset, Liability and Null. According to this the derivative instruments can change the phases over time without losing its identity by switching from the Phase Null, i.e. an off-balance position, into an on-balance position either to the Phase Asset or the Phase Liability and so on.

Examples:

- Stock exchange (e.g. forward contracts)
- Claims can only change from off- to on-balance not from asset to liability.



1.Fin-security-master: one marketplace at a certain time where the price changes.
In the domain where (the stocks exchange) and price goes up and down.
Deriv-Instr-master = Information about the contract (sign the option at certain date with interest rate,...)
2. Deriv-Instr-Trans: it saves the data calculated for the option with specific time and fair value (asset,liability)
3. asset, liability, off balance: they represent the phases, where the Deriv-Instr-Trans can be asset, liability or off balance
Ä

Collective: This is the type of trade (Short Call, Long Call, …) and based on this two economic resources are exchanged
MemberOf: A economic resource can be an Asset, Liability, Equity, Claim in differen times.
Master Classes: Commitment, Debit Commitment, Credit Commitment, Balanced Reciprocity
Transactional Classes: Economic Event, Debit Event, Credit Event, Balanced Duality
Debit & Credit Event: ????

Black/Scholes Call Option Pricing – Primer (1/3)

- Black/Scholes „formula“ for pricing (fair value = FV) European call options (= C) at time of initiating the contract (t = 0)

$$FV_{C,0} = \underbrace{+P_{A,0} \cdot N(d_{1,0})}_{fin.Asset(left\ leg)} \underbrace{- X_{0,T} \cdot N(d_{2,0}) \cdot \exp(-r_{0,T} \cdot T_{0,T})}_{fin.Liab.(right\ leg)}$$

mit

$$d_{1,0} = \frac{\ln(P_{A,0} / X_{0,T}) + (r_{0,T} + {\sigma_{0,T}}^2 / 2) \cdot T_{0,T}}{\sigma_{0,T} \cdot \sqrt{T_{0,T}}}$$

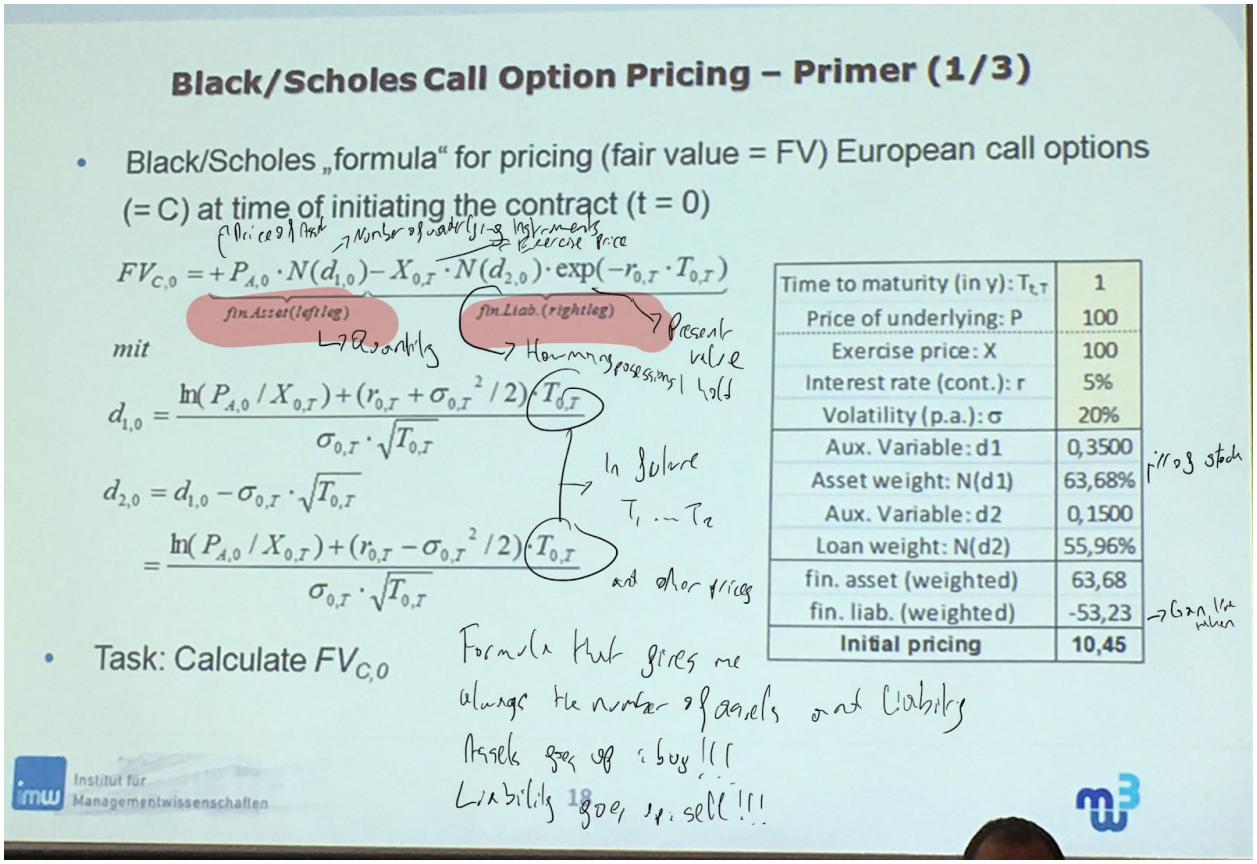
$$d_{2,0} = d_{1,0} - \sigma_{0,T} \cdot \sqrt{T_{0,T}}$$

$$= \frac{\ln(P_{A,0} / X_{0,T}) + (r_{0,T} - {\sigma_{0,T}}^2 / 2) \cdot T_{0,T}}{\sigma_{0,T} \cdot \sqrt{T_{0,T}}}$$

Time to maturity (in y): $T_{t,T}$	1
Price of underlying: P	100
Exercise price: X	100
Interest rate (cont.): r	5%
Volatility (p.a.): σ	20%
Aux. Variable: d1	0,3500
Asset weight: N(d1)	63,68%
Aux. Variable: d2	0,1500
Loan weight: N(d2)	55,96%
fin. asset (weighted)	63,68
fin. liab. (weighted)	-53,23
Initial pricing	**10,45**

- Task: Calculate $FV_{C,0}$

P = Aktueller Aktienpreis



X = Aktienpreis zu dem ich in Zukunft die Aktie kaufen werde (Im Vertrag ausgehandelt)
R = Riskless Interest, wenn ich das Geld Risikofrei investiere, bekomme ich r% zinsen
Klein Sigma = Kursschwankungen
N(d1) = Wahrscheinlichkeit, dass der Call executiert wird
N(d2) = Gegenwahrscheinlichkeit zu N(d1), dass der Call nicht exekutiert wird.
exp(-r*t) = Nominalwert, So viel ist X in Zukunft wert.

In english: N(d1,0) is the probability that the calls gets executed. So therefore N(d2,0) is the counter probability that the call will not be executed. All clear? :)

Resultat = So viel ist der Call fairer Weise wert.

Black/Scholes Call Option Pricing – Primer (2/3)

- European call option pricing: Calculations

European Call (Derivative FI)	t0		t1		t2	
Cash Flow	-10,45					20,00
Time to maturity (in y): $T_{t,T}$	1		0,50		0,00	
Price of underlying: P	100,00		100,00		120,00	
Exercise price: X						100,00
Interest rate (cont.): r	5,00%		5,00%			
Volatility (p.a.): σ	20,00%		20,00%			
Aux. Variable: d1	0,3500		0,2475			
Asset weight: N(d1)	63,68%		59,77%		100,00%	
Aux. Variable: d2		0,1500		0,1061		
Loan weight: N(d2)		55,96%		54,22%		100,00%
Hedge portfolio: PF	fin. Asset	fin. Liab.	fin. Asset	fin. Liab.	fin. Asset	fin. Liab.
fin. asset (weighted)	63,68		59,77		120,00	
fin. liab. (weighted)		-53,23		-52,88		-100,00
Initial pricing	10,45					
Subsequent pricing			6,89		20,00	
Price change			-3,56		13,11	

Black/Scholes Call Option Pricing – Primer (3/3)

- European call options: Black/Scholes formula is more than simply a formula => it specifies the hedge portfolio replication „policy“ for no-arbitrage pricing

Exercise price	100,00										
Interest rate (cont.)	5,00%										
Volatility (p.a.)	20,00%										
Initial Pricing			Nd1t	Value			Present Value	Fair Value			
Time to maturity (y)	Stock price	d1,t	N(d1,t)	Asset value	d2,t	N(d2,t)	Liab. value	**PF value**			
1,00	100,00	0,3500	63,68%	63,68	0,1500	55,96%	-53,23	**10,45**			
Subsequent pricing			values before transaction (b.t.)						values after transaction (a.t.)		
Time to maturity (y)	Stock price	N(d1,t-1)	A. value b.t.	L. value b.t.	**PF value b.t.**	d1,t	N(d1,t)	ΔN(d1,t-1)	A. value a.t.	L. value a.t.	**PF value a.t.**
0,75	100,00	63,68%	63,68	-53,90	9,78	0,3031	61,91%	-1,77%	61,91	-52,13	9,78
0,50	100,00	61,91%	61,91	-52,78	**9,13**	0,2475	59,77%	-2,14%	59,77	-50,65	**9,13**
0,25	120,00	59,77%	71,73	-51,29	20,44	1,9982	97,72%	37,94%	117,26	-96,82	20,44
0,00	120,00	97,72%	117,26	-98,03	**19,23**	0,0000	0,00%	0,00%	0,00	0,00	0,00

Black/Scholes and Merton introduced the no-arbitrage pricing theory in 1973. According to them, the **no-arbitrage price, the fair-value,** is calculated from the **asset value and the present value of the liability.** This **value changes over the lifetime of a call option**, due to the dependence of the asset weight evolving over time. For this reason, it makes the

composition dynamic. Indeed, the dynamic hedge portfolio peculiarities are now part of the Collective class Derivative Instrument, the MemberOf relationship between Collective class Derivative Instrument and Kind class Economic Resource and the Formal relationship in-/outflow (out-/inflow) between Kind class Economic Resource and SubKind class Debit Event (Credit Event)[1].

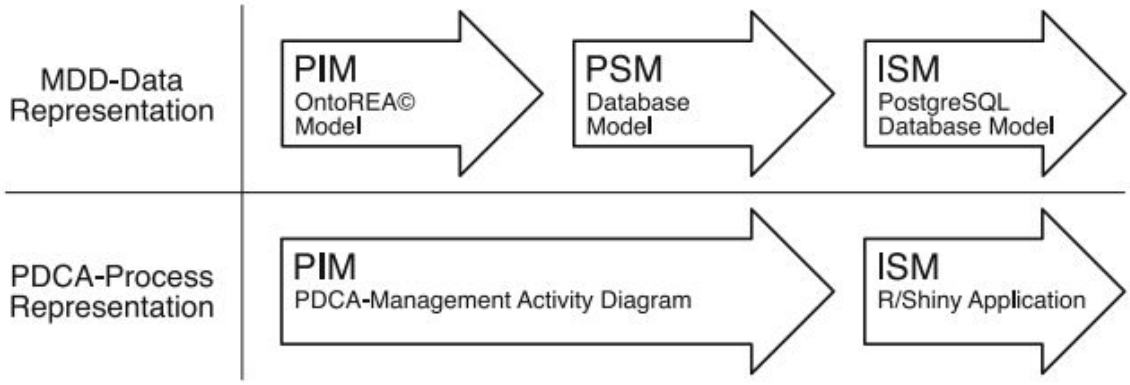
[1] OntoREA© Accounting and Finance Model: Hedge Portfolio Representation of Derivatives: 11th IFIP WG 8.1. Working Conference, PoEM 2018, Vienna, Austria, October 31 – November 2, 2018, Proceedings

Paper 5 - Please feel free to edit & comment

OntoREA© model holds true for static and dynamic hedge portfolio compositions.

Steps of a **retroactive design science methodological** (DSRM) perspective within the **model driven software development** (MDD) context

1. OntoREA˝ model serves as **platform independent (PIM) model.**
2. direct translation into a **platform specific (PSM) model** (in Paper SQL + OCL)
3. **implementation specific (ISM) model** (in Paper RStudio/Shiny)



MDD - model driven development, create a OntoREA model which describes the constraints and entities, try to generate a database model out of the OntoUML diagram (model language is more expressive than UML), convert the database model to a postgreSQL database model to save the transactional and master data in the database

PDCA - Process: Plan-Do-Check-Act Model => referred in the 3-tier architecture, dynamic infrastructure (plan activity, do something/check, act = conditional deliverties (?))

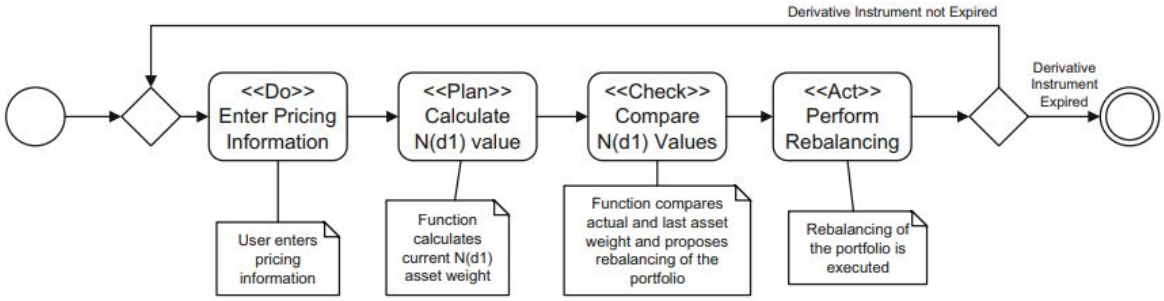
DSRM Methodology (applied in Paper)

(1) Problem identification and motivation,
(2) definition of the objectives of the solution
(3) design and development
(4) demonstration
(5) evaluation
(6) Communication

To put it in a nutshell the OntoREA© accounting and finance model has included the notion of static and dynamic hedge portfolio and demonstrated that it's possible to design an application.

Replicating Derivatives in R-Shiny-ERM: MDD

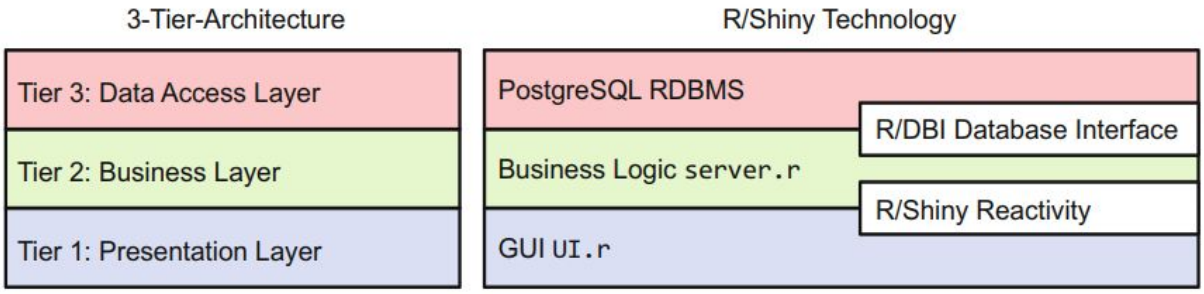
The key for implementing the PDCA management process shown below is the reactivity technology representation of the dynamic replication policy.



After having specified the PIM process model for the dynamic replication policy in form of the PDCA management activity diagram it can be translated into the R/Shiny ISM process model.

- First the <<DO>> prompts the user to enter the current pricing information
- In the next step, <<PLAN>>, the current assets get calculated and displayed
- Now in <<CHECK>> the actual values where compared against the last observation of the pricing and the difference is calculated
- And at last in the <<ACT>> activity the action is executed and the according data is inserted into the PostgreSQL database

Implementation of the 3-tier-architecture in R/Shiny



The code programmed in R/shiny is the implementation of the 3-Tier-Architecture that facilitate the extraction of the derivatives data (Data Access Layer), manage the conditions of the derivative and its calculus (Business Layer) and finally shows the result to the user (Presentation Layer). The code also allows to record new data related with new derivatives taking the way back from the first layer until the third layer. In the code that was provided we

can see the process how it works, First it create the functions required for the calculus and the management, also the communication with the user representing the reactivity in R/Shiny. After it shows there is not register of derivatives but it continues with a register of a derivative. With the new register the code allows to calculate fair value, calculate the N(d1) -Plan- , compare the values -Check-, and finally it shows the final result to perform the balancing -Act-. This process is reiterative until the derivative instrument expired. The importance of this process is located in the fact it works like technical instrument to support decisions also to account the behavior of the derivative instruments looking to make them as asset (profits) for the stakeholders.

Necessity of Equity_Change in the database PSM model

The introduction of the equity resource type is needed for capturing the revenues and expenses that occur by executing the dynamic replication policy over time. This is also shown in the image below.

	Risky Income Change (A)		Fixed Income Change (L)		Equity Change (E)	
	Value Change		Present Value Change			
Date	Debit	Credit	Debit	Credit	Debit Value	Credit Value
01.01.	63.68			53.23		10.45
31.03.		1.77	1.10		0.67	
30.06.		2.14	1.48		0.66	
30.09.	57.49			46.17	0.63	11.95
31.12.	0.00			1.21	1.21	

Demonstrate the concept of reactivity with Pascal Triangle.

The reacting programming model involves basically three elements: (1) Reactive Source, (2) Reactive Conductor and (3) Reactive Endpoint. To demonstrate this concept we decided to use the Pascal Triangle involving this concepts.

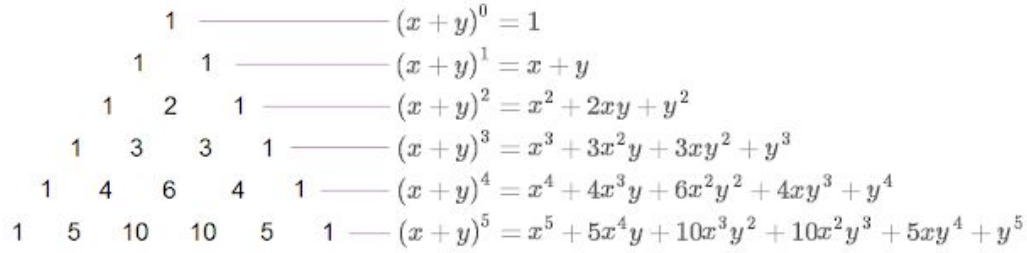
The Pascal triangle is an configuration that allows to obtain the binomial coefficients. Using the concepts of reactivity we will define the step of the calculation as follows.

Reactive Source: The number of rows (grade of the binomio) that is an input indicated for the user.

Reactive Conductor: We use the condition. If the number of rows is equal to 0 then, the pascal triangle will show the first level, if it is 1 then show the basic form of the binomio (a+b) but if is higher than 1 (2, 3, …..) do the calculus and build the triangle.

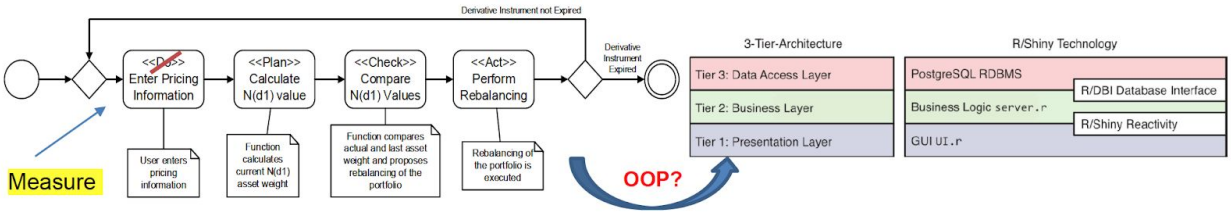
Reactive Endpoint : For this step we will ask to the application to build the triangle and show the factorization form and formula.

The application will interact with the user through a slide that allows to select between the minimum number of rows (0) and the maximum (defined for the programmer)



Friend of mine gave me a good explanation:
Guys!!! the principle of reactivity is just a simple way to design interactive application with Shiny for R software
it involves basically three steps:
(1) reactive sources -> the incentive for the app;
(2) Reactive conductor->intermediate calculation or logic rules; and
(3) Reactive endpoint->the consequence or output
applying in pascal triangle we can indicate in
(1) the grade of the binomio you want to factorize, then
(2) do the calculations and
(3) shows the result.
The app in interactive then every time you change the grade of the binomio the output will change with no necessity to run a code or something

Group Work: MDD – Case: PDCA-Modeling of Replication Policy

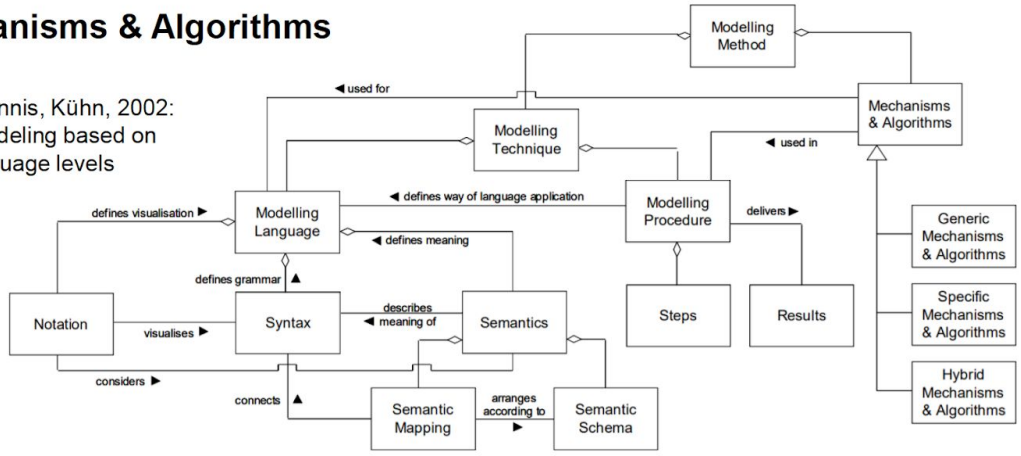


- **Case**: Modeling the option replication policy as PDCA-management process and translating it with OOP methodology into the R/Shiny 3-tier architecture
  - How are the Measure, Plan, Check and Act activities implemented?
  - Which information is flowing between the activities and how is the flow impl.?
  - Which information is entered/presented in the Presentation layer?
  - Which information is processed in the Business layer?
  - Which information is retrieved/stored from the Data Access layer?

Paper 6 - Please feel free to edit & comment

- Karagiannis/Kühn: **Modelling Method** consists of
- 1) **Modelling Technique**, which is composed of **Modelling Language** with Notation, Syntax and Semantics, and **Modelling Procedure**, and
- 2) **Mechanisms & Algorithms**

Karagiannis, Kühn, 2002: Metamodeling based on language levels



While business requirements are rapidly changing and in need of complex applications, the *state-of-the-art* in modelling of organisations is still based on *fixed* metamodels. In addition, business architectures depend highly on the branches under consideration, and some major requirements to an

enterprise modelling platform are flexibility, adaptability, and openness for the integration of models based on different modelling paradigms

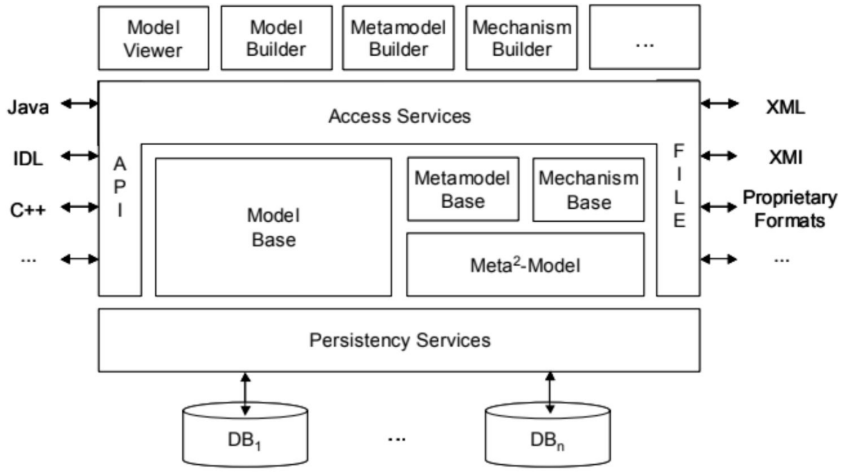


Fig. 4: Generic architecture of metamodelling platforms

(1) **Modelling methods** consist of a

1. modelling technique, which is composed of a **modelling language** consisting of syntax, semantics and notation, and a **modelling procedure** which describes the steps applying the modelling language to create results, i.e. models. Metamodels are created by metamodeling languages, defined by meta-metamodels or meta^2-models (Geisler, Klar & Pons 1998; Strahringer 1996). The authors emphasize the importance of clear semantics in order to find a useful level of abstraction for 'finishing' the modelling hierarchy.

2. mechanisms & algorithms, working on the models describes by the modelling language.

(2) The authors suggest a generic architecture for metamodeling platforms, emphasizing their component-based, distributable and scalable form. The main components are:

The persistency services manage the storage of all model and metamodel information.

The meta^2-model, which is the central part of the architecture, provides the basic concepts to create metamodels and mechanisms ('classes', 'relations', 'attributes', etc.).

The metamodel base contains all information about the metamodels currently managed by the modelling platform.

The mechanism base contains information about functionalities to be applied to models and metamodels.

The model base contains all models based on the metamodels. It communicates with the metamodel base.

Access services provide file-based and online interfaces to the different types of bases.

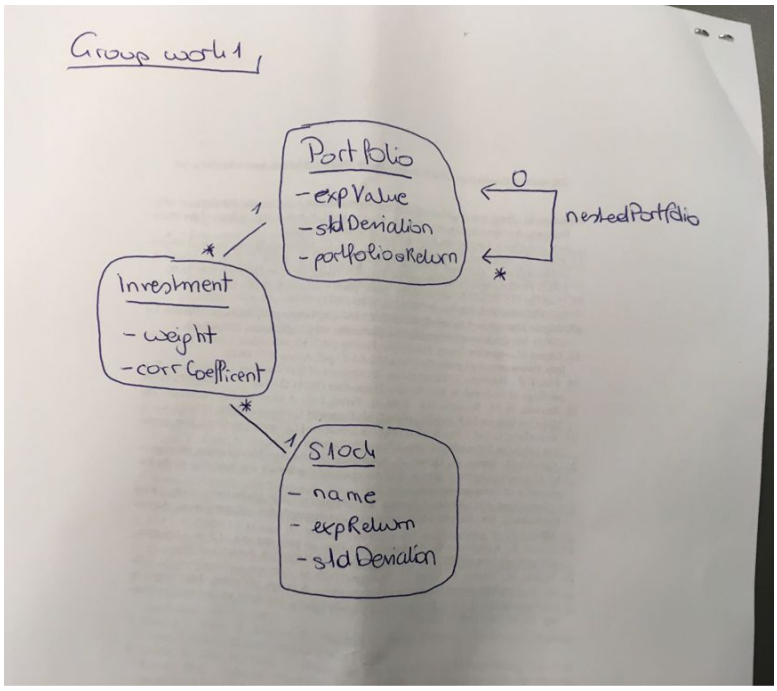
Different **viewer and builder components** support the usage and maintenance on top of the access services.

(3) Finally, the authors emphasise major advantages of flexible metamodel approaches, like savings in time and costs in application development, increased quality of delivered solutions, and enhanced acceptance because of directly mapping the domain under consideration.

Group Work 1: Meta-Modeling – Stock Portfolio Management (1/2)

- **Case**: In your function as junior treasurer your are responsible for the management of stock portfolios. With the acquired meta-modeling knowledge you are keen to apply it to your daily business.
- For modeling a stock portfolio you identified the following ingredients:
  - The portfolio can hold **"one to many"** stocks.
  - Each stock has a **name** and two attributes, i.e. the **expected return** and the return's **standard deviation**.
  - The portfolio stores the **portfolio weights** invested in each stock, the **correlation coefficients** among all pairs of stocks, and the **expected value** as well as the **standard deviation of the portfolio return**.
  - Finally, the portfolio can be nested within other financial portfolios.
- Design the meta-model that contains all stock portfolio ingredients.

Goo



Group Work 1: Meta-Modeling – Stock Portfolio Management (2/2)

- Portfolio return: Calculate the expected return and the standard deviation.
- Does your stock portfolio meta-model provide all information for these calculations?

	Stock A	Stock B	Stock C
Price	100	85	65
Expected Return	6%	11%	15%
Std. Deviation σ	0.1	0.25	0.4
Portfolio weight w	*0.4*	*0.34*	*0.26*

	Stock A, B	Stock B,C	Stock A,C
Correlation coeff. ρ	0.1	0.2	-0.8

Portfolio Return

$$= \sum w_i \times E(R_i)$$

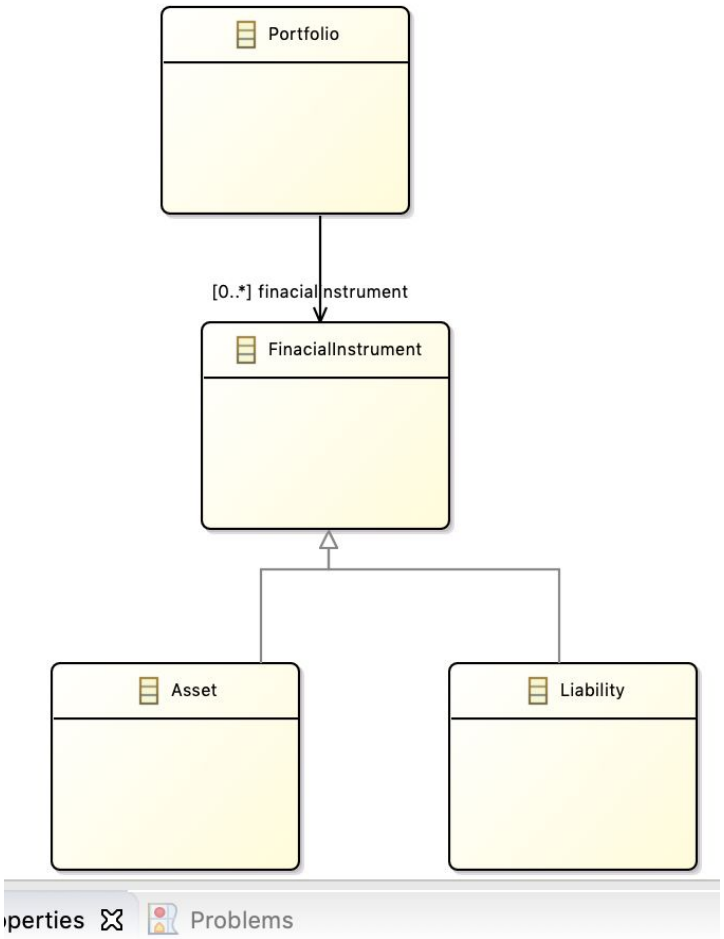
Portfolio Variance

$$= w_A\sigma_A^2 + w_B\sigma_B^2 + w_C\sigma_C^2 + 2w_Aw_B\rho_{A,B}\sigma_A\sigma_B + 2w_Aw_C\rho_{A,C}\sigma_A\sigma_C + 2w_Bw_C\rho_{B,C}\sigma_B\sigma_C$$

(changed MULTIPLICATION to SUM as noted by lecturer)
Portfolio return: 0.10
Portfolio variance: 0.06441

Group Work 2: Meta-Modeling – Asset/Liability-Management

- **Case**: After successfully working in the stock portfolio management you were promoted. Now you are responsible for the Asset/Liability-Management of your customers portfolios.
- In Asset/Liability portfolios consists of financial instruments that can not only be assets, but also liabilities.
- Create a meta-model satisfying the new requirement
- Design a meta-model that adequately represents the asset/liability portfolio



Group Work 3: BUS & MGT Domain: Meta-Modeling and DSL

- **Case**: After the demonstration of functionality in ADOxx you are reminded to the concepts of the paper from Karagiannis & Kühn (2002).
- What are the main concepts of ADOxx and what do they consist of?
- Map the ADOxx concepts to the concepts that underlie Karagiannis & Kühn's definition of a modeling method.

Modeling Method (KK2002)
Modeling Language
Modeling Procedure
Algorithms/Mechanisms