(Big) Refactoring de modèles UML

HAI913I - 2021

Outline

Motivation

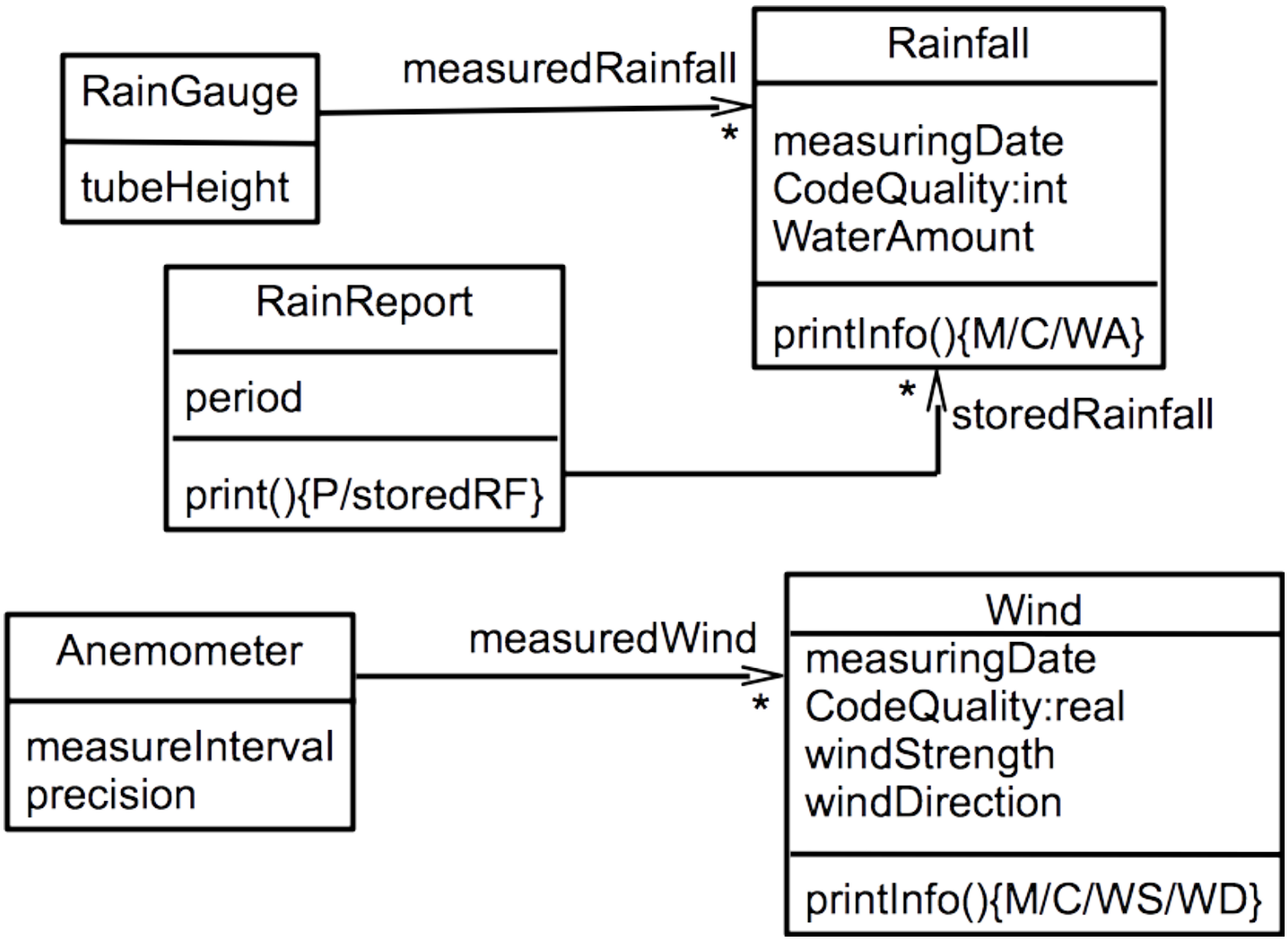
Intérêt des modèles de classes

- Capturer la connaissance de domaine et la représenter
- Mettre en lumière la classification
- Favoriser la réutilisation

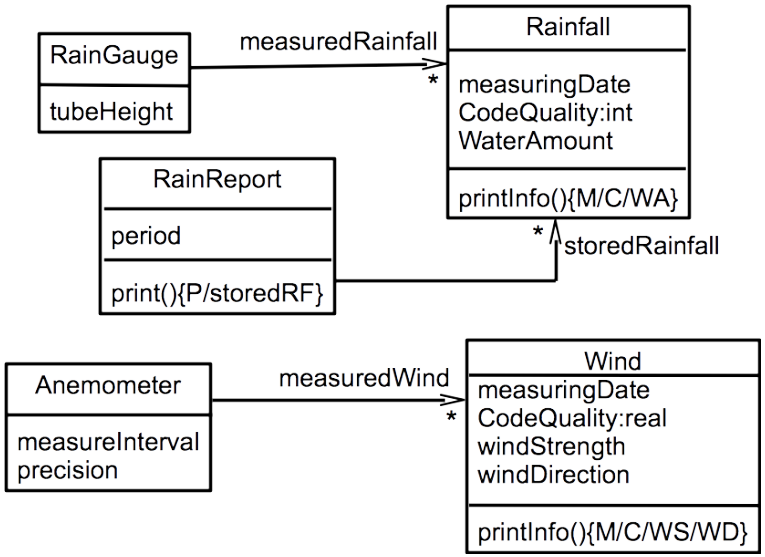
Un modèle en forme normale

- Sans redondance
- Intégrant toutes les abstractions
- Intégrant toutes les relations de spécialisation
- Structure la plus compacte possible

Modèle de classes initial

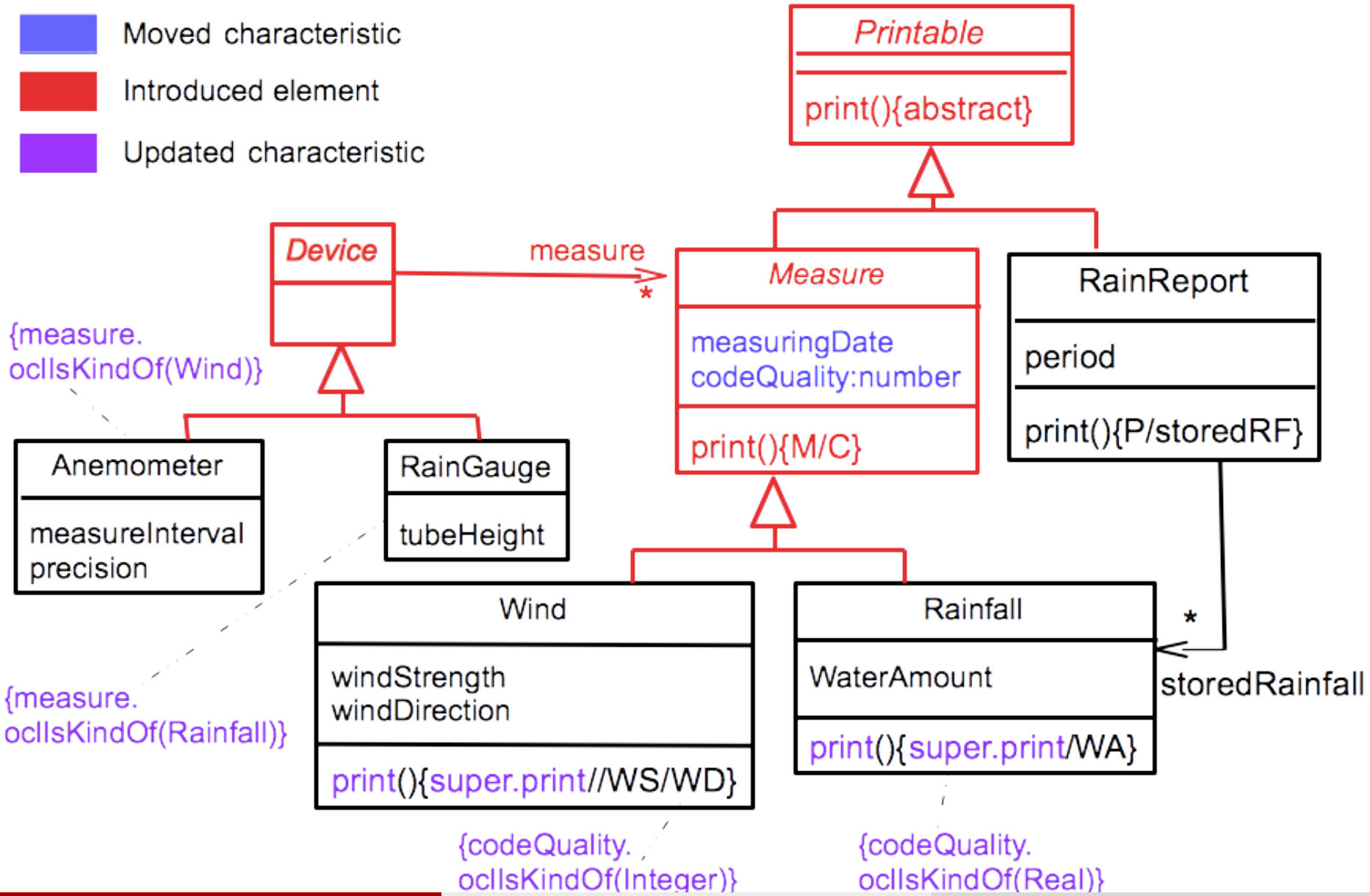


Modèle de classes initial



- `measuringDate` et `codeQuality` sont répétées dans `Rainfall` et `Wind` (notion de *Measure*)
- `RainGauge` et `Anemometer` sont connectées via `measured...` (notion de *Device*)
- les méthodes `print` partagent du comportement (M/C = print measuringDate / codeQuality).

Modèle de classes final



Etapes dans le processus

- Formal Concept Analysis *(Ganter and Wille 1999)*
  Galois lattices *(Barbut and Monjardet 1970)*
  - Flat characteristics *(Godin et al., 1993)*
  - Hierarchical characteristics *(Godin et al., 1993)*
- Relational Concept Analysis *(Rouane Hacène et al. 2013)*
  - Reified characteristics *(Roume et al. 2004)*
  - Clustered reified characteristics *(unpublished)*

Outline

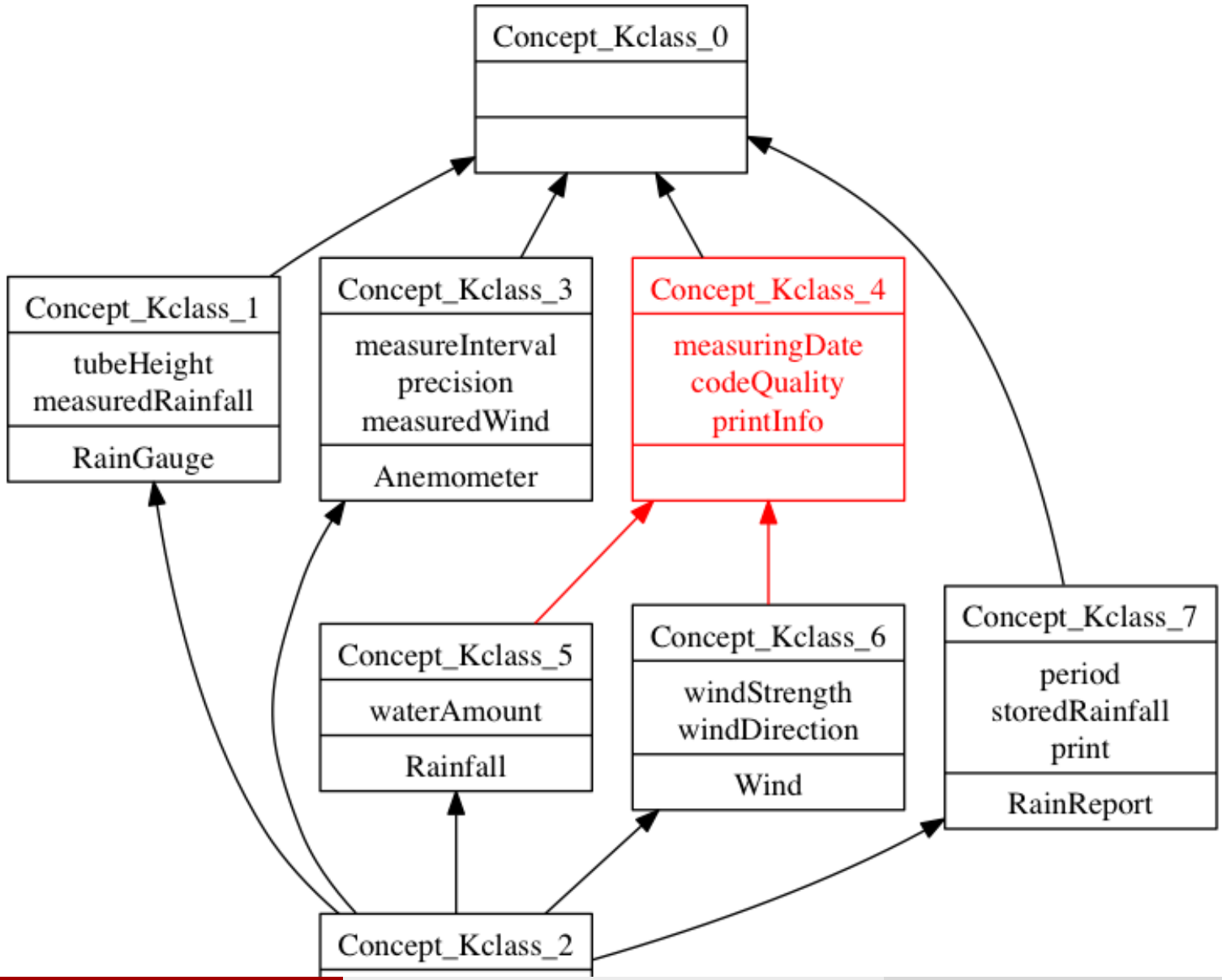
Flat characteristics: Formal context (Godin and Mili, 1993)

Kclass	tubeHeight	measureInterval	precision	measuringDate	codeQuality	waterAmount	windStrength	windDirection	period	measuredRainfall	measuredWind	storedRainfall	print	printInfo
RainG.	×									×				
Anem.		×	×								×			
Rainf.				×	×	×								×
Wind				×	×		×	×						×
RainR.									×			×	×	

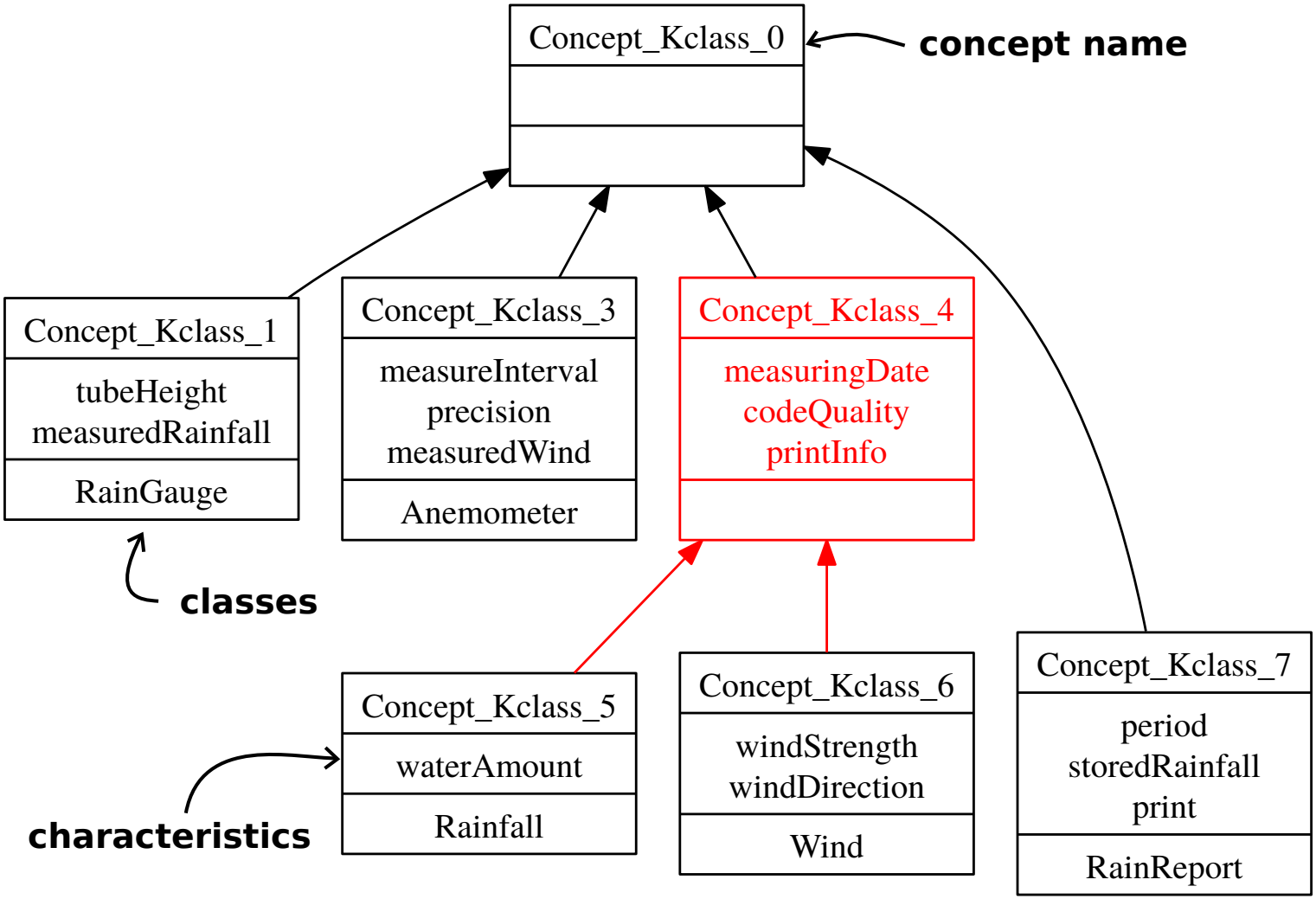
Flat characteristics: Concept

Kclass	tubeHeight	measureInterval	precision	measuringDate	codeQuality	waterAmount	windStrength	windDirection	period	measuredRainfall	measuredWind	storedRainfall	print	printInfo
RainG.	×									×				
Anem.		×	×								×			
Rainf.				×	×	×								×
Wind				×	×		×	×						×
RainR.									×			×	×	

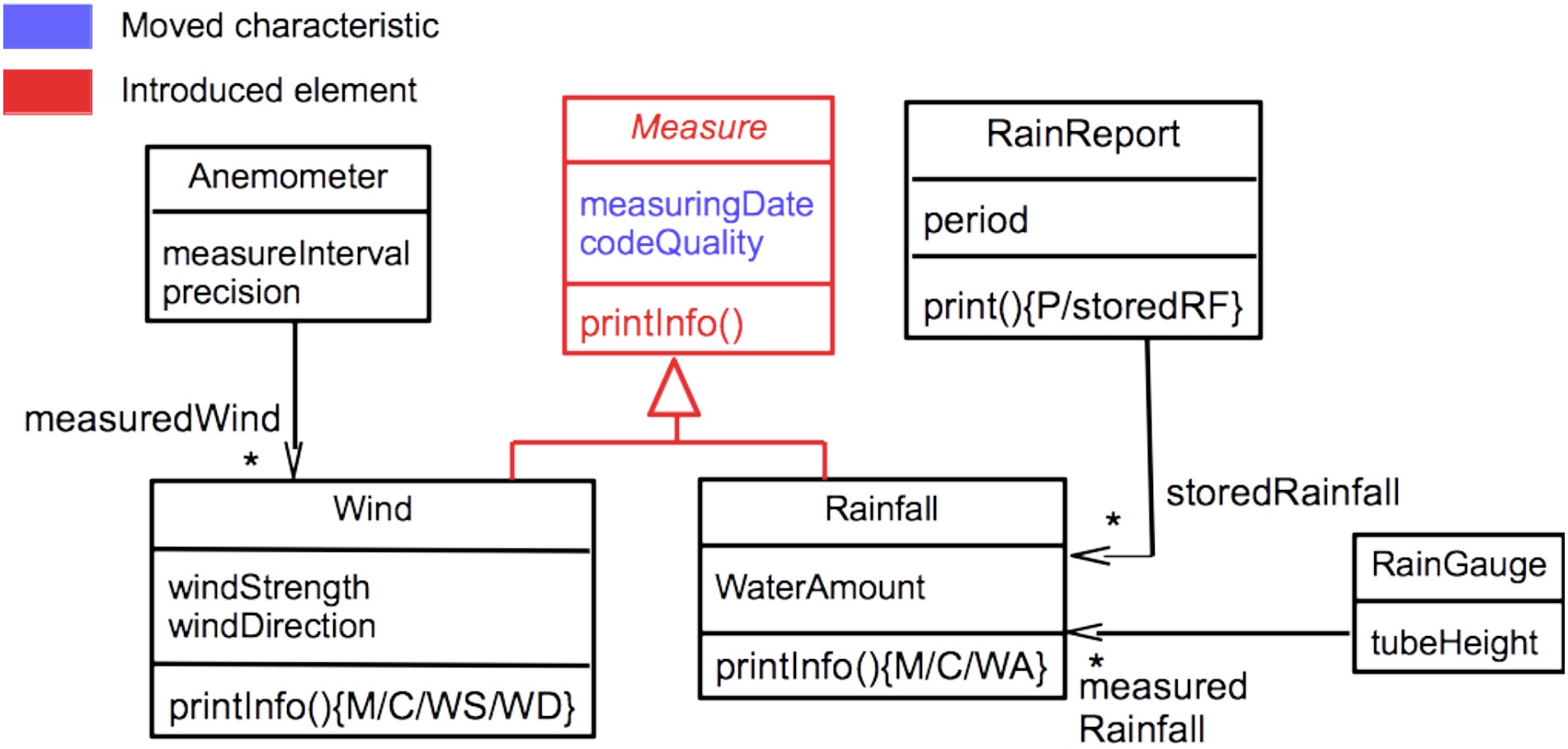
Flat characteristics: concept lattice



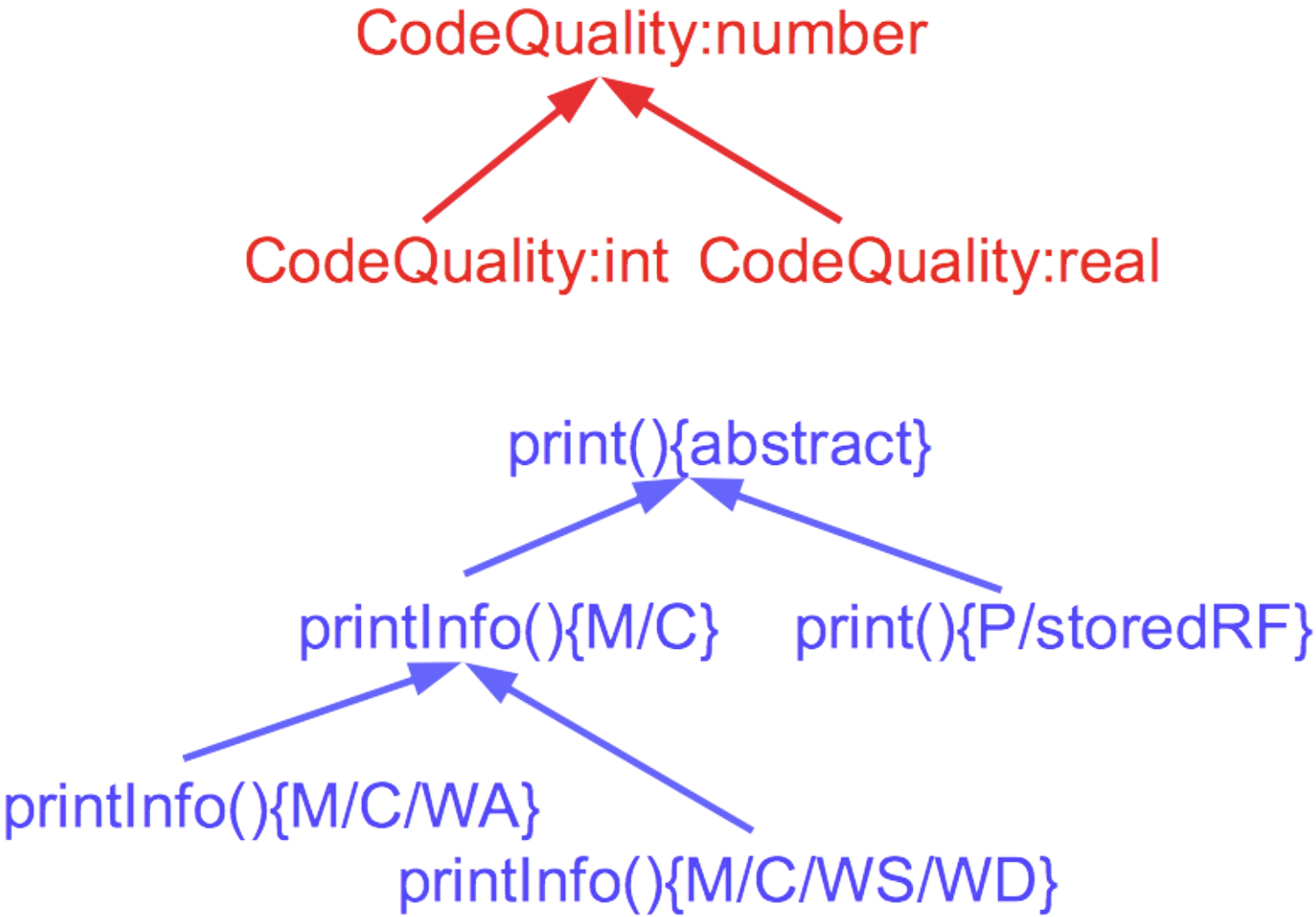
Flat characteristics: concept lattice (without bottom)



Flat characteristics: revisited model



Hierarchical characteristics (Godin and Mili, 1993)



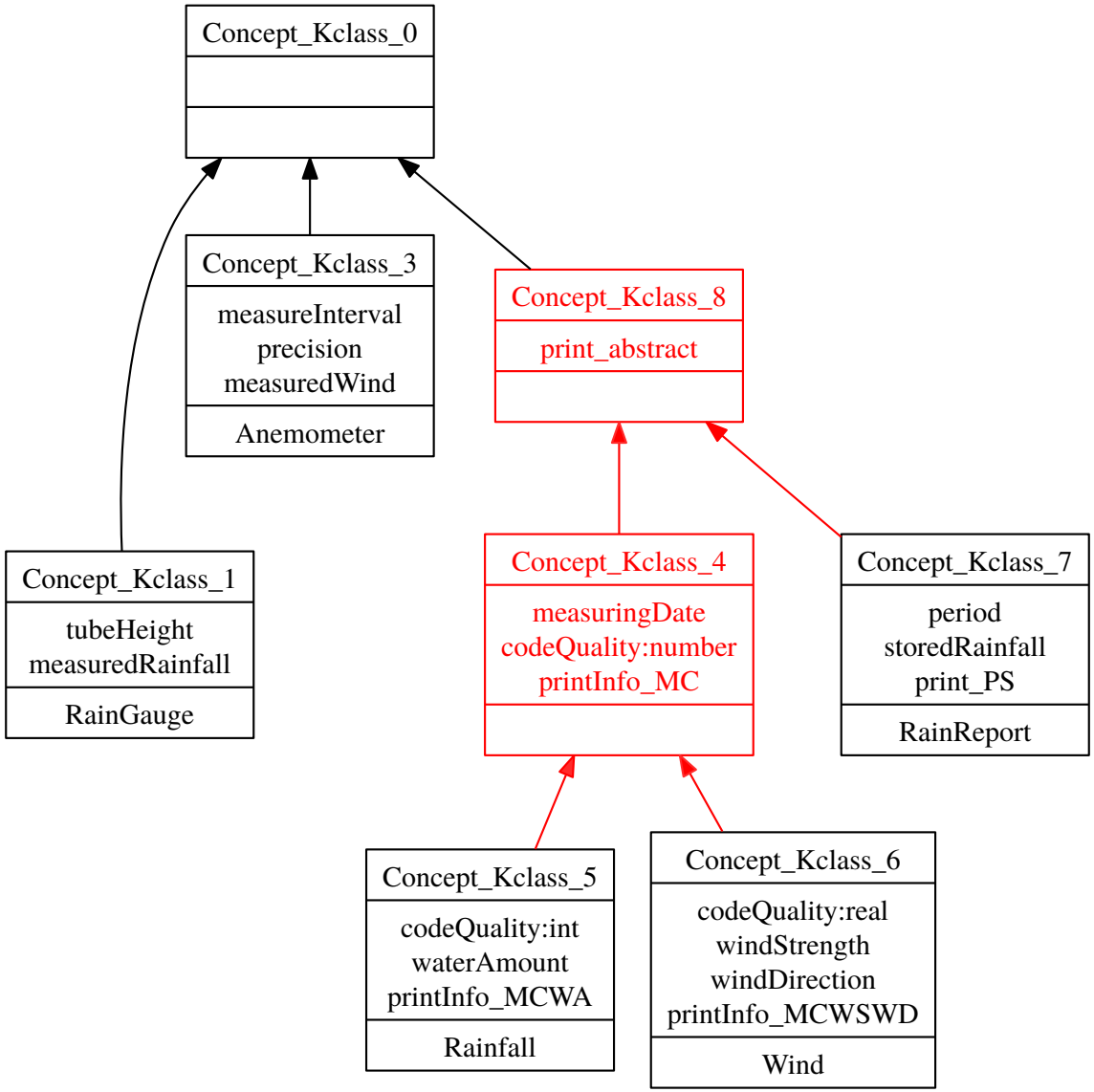
Hierarchical characteristics: Formal context (part 1)

Kclass	tubeHeight	measureInterval	precision	measuringDate	codeQuality:number	codeQuality:int	codeQuality:real	waterAmount	windStrength	windDirection	period	measuredRainfall	measuredWind
RainGauge	×											×	
Anemometer		×	×										×
Rainfall				×	×	×		×					
Wind				×	×		×		×	×			
RainReport											×		

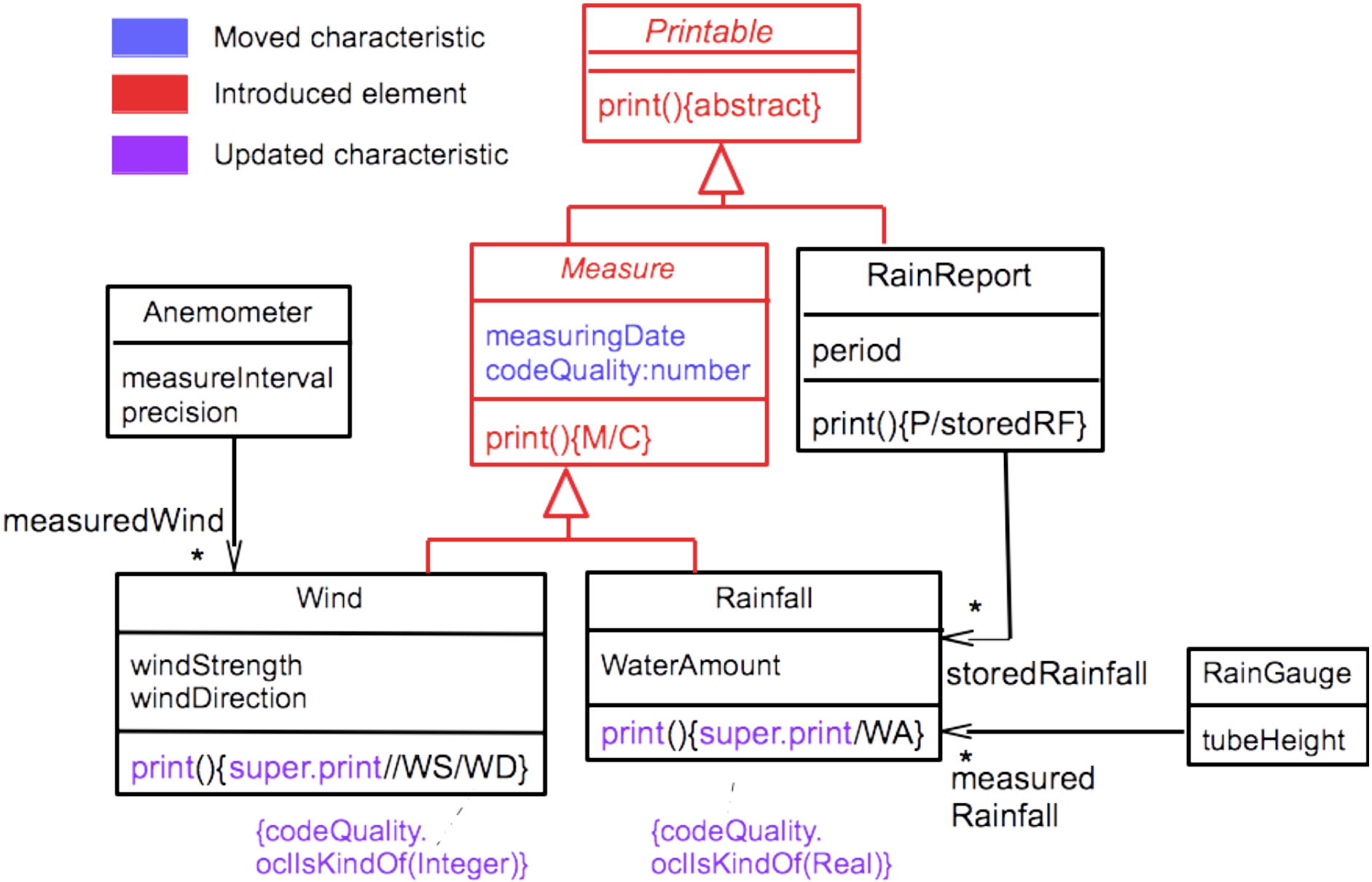
Hierarchical characteristics: Formal context (part 2)

Kclass	measuredWind	storedRainfall	print_abstract	printInfo_MC	printInfo_MCWA	printInfo_MCWSWD	print_PS
RainGauge							
Anemometer	×						
Rainfall			×	×	×		
Wind			×	×		×	
RainReport		×	×				×

Hierarchical characteristics: Concept Lattice (sans bottom)



Hierarchical characteristics: revisited model



Outline

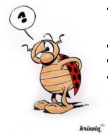
Parenthèse sur : Relational Concept Analysis

- Prendre en compte différentes catégories d'objets et des liens entre ces objets
- Principe :
  - Un modèle basé sur le modèle entité-relation
  - Entités : contextes formels
  - Relations : contextes relationnels
  - Intégration des relations entre objets sous forme d'attributs *relationnels*
  - Un processus itératif
- RCA produit un ensemble de treillis interconnectés (un treillis par contexte formel)

Un contexte formel décrivant des drones

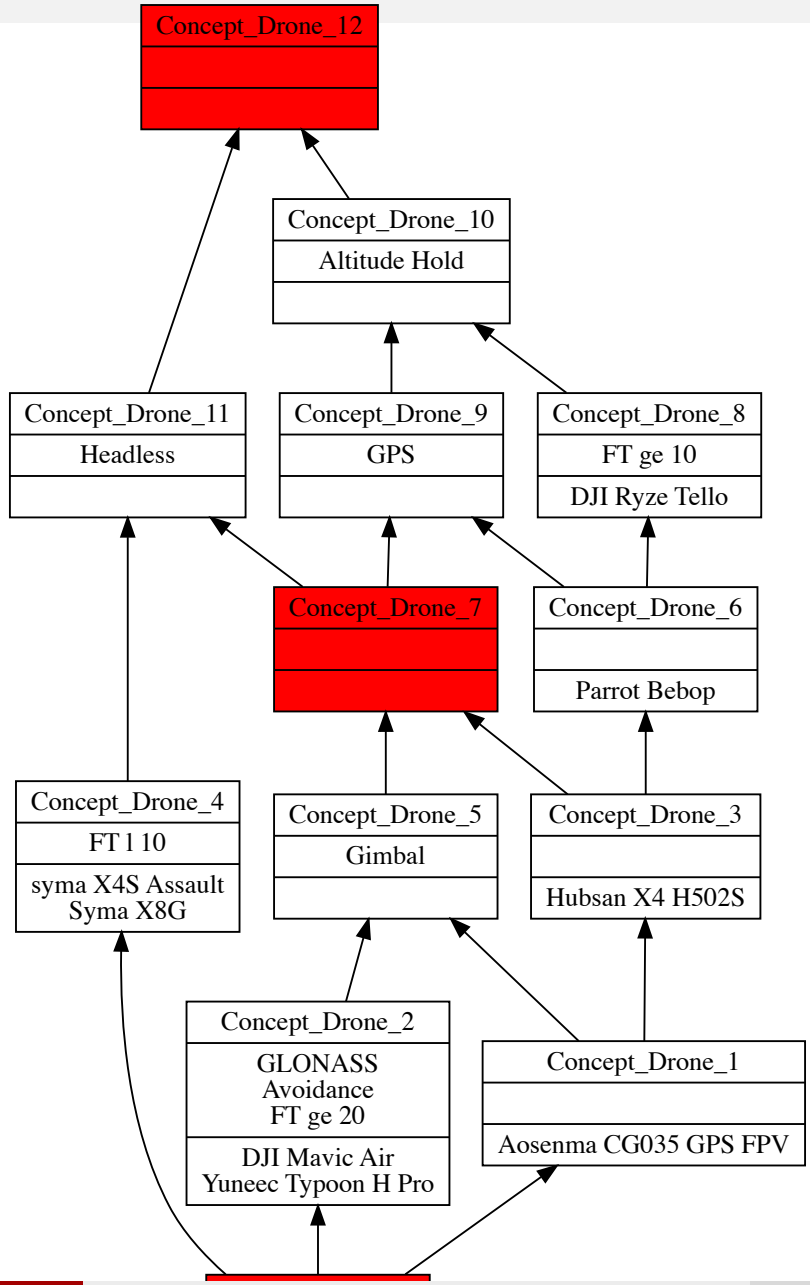
Drone	Gimbal	GPS	GLONASS	Avoidance	Headless	Altitude Hold	FT I 10	FT ge 10	FT ge 20
Syma X4S Assault 					×		×		
Syma X8G 					×		×		
Parrot Bebop 		×				×		×	
DJI Ryze Tello 						×		×	
Hubsan X4 H502S 		×			×	×		×	
Aosenma CG035 GPS FPV 	×	×			×	×		×	
DJI Mavic Air 	×	×	×	×	×	×			×
Yuneec Typoon H Pro 	×	×	×	×	×	×			×

https://www.thedronechart.com

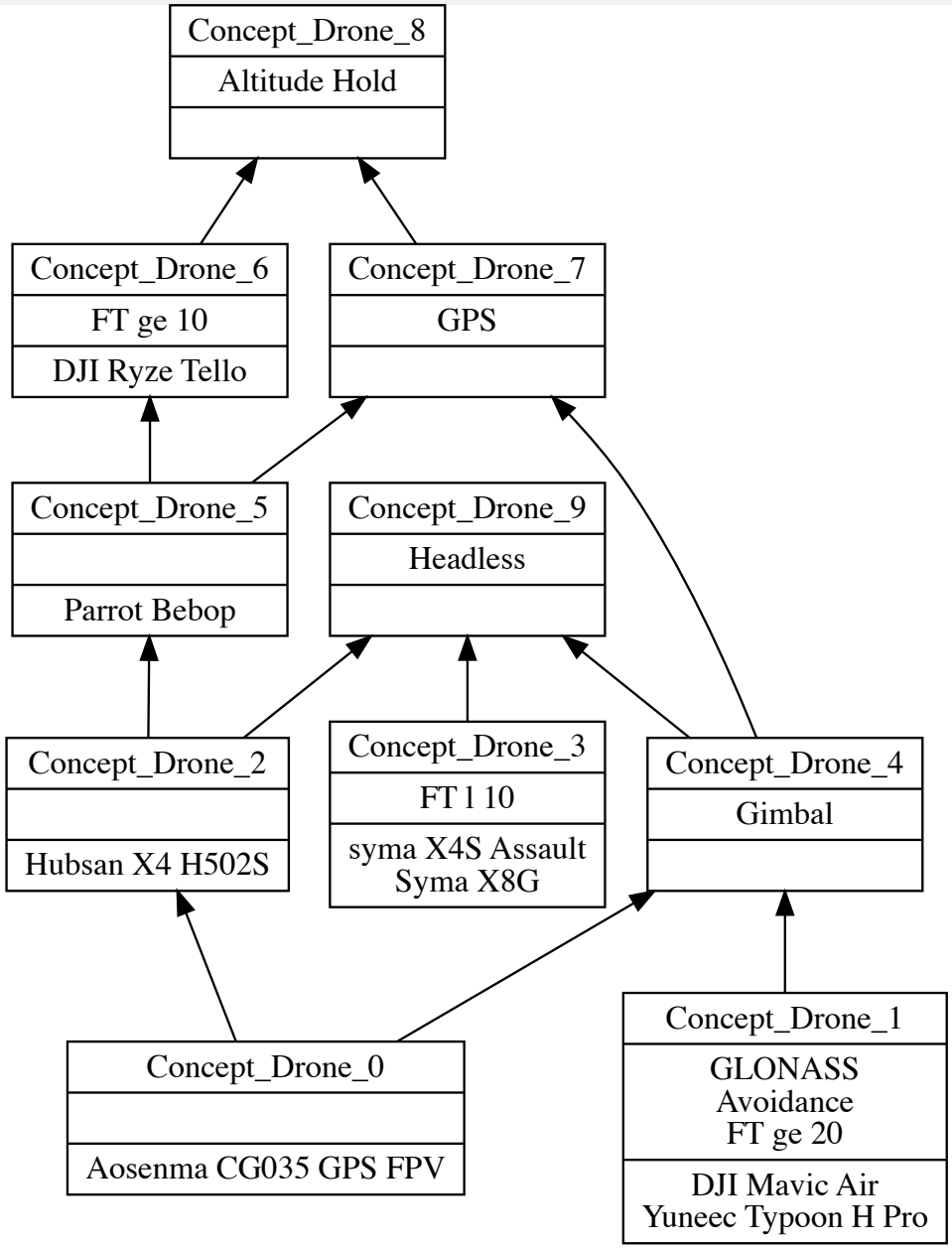


Hum, typo, here. Typoon in World of Warcraft? Teaspoon?

Le treillis / l'AOC-poset des drones



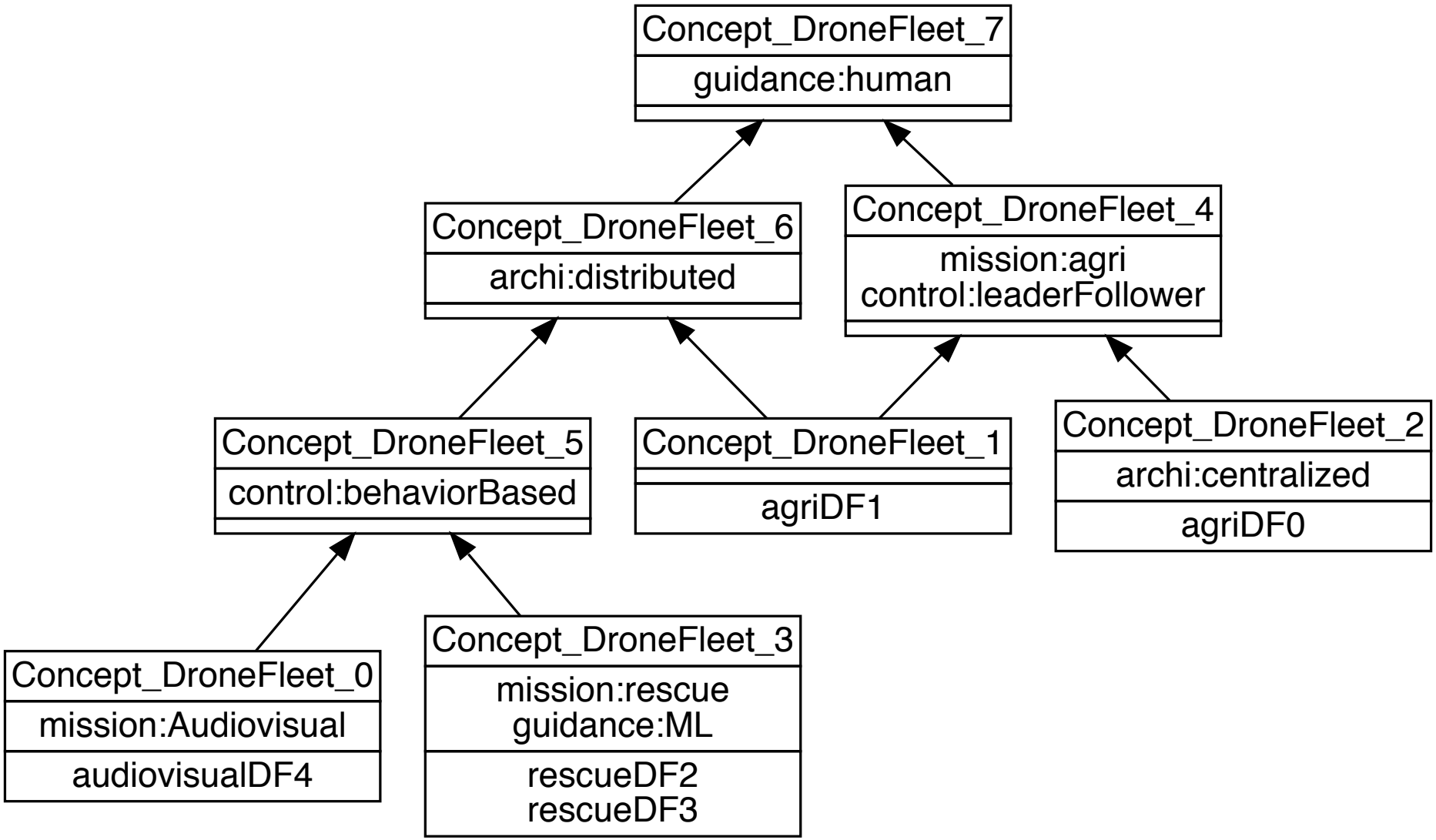
Le treillis / l'AOC-poset des drones



Un contexte formel pour des flottes de drones

DroneFleet	mission:agri	mission:rescue	mission:Audiovisual	archi:centralized	archi:distributed	guidance:human	guidance:ML	control:leaderFollower	control:behaviorBased
agriDF0	×			×		×		×	
agriDF1	×				×	×		×	
rescueDF2		×			×	×	×		×
rescueDF3		×			×	×	×		×
audiovisualDF4			×		×	×			×

L'AOC-poset des flottes de drones



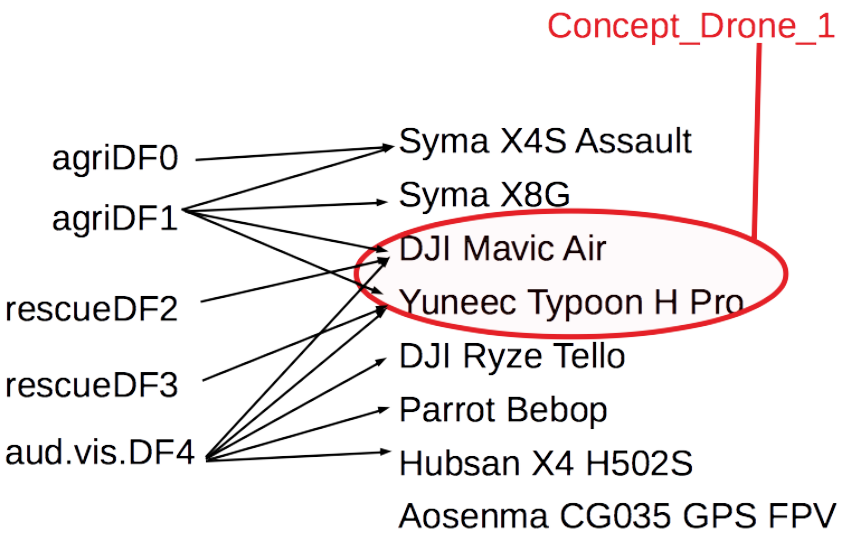
Le contexte relationnel liant les flottes de drones aux drones qu'elles contiennent

DroneFleet	Syma X4S Assault	Syma X8G	Parrot Bebop	DJI Ryze Tello	Hubsan X4 H502S	Aosenma CG035 GPS FPV	DJI Mavic Air	Yuneec Typoon H Pro
agriDF0	x							
agriDF1	x	x					x	x
rescueDF2							x	
rescueDF3								x
audiovisualDF4			x	x	x		x	x

Scaling quantifiers

`rescueDF2` and `rescueDF3` do not share concrete drone types, but they share the fact that all their drones with `GLONASS`, `GPS`, `FT` $\geq$ 20, etc.

Relational attribute: $\exists\forall contains(Concept\_Drone\_1)$

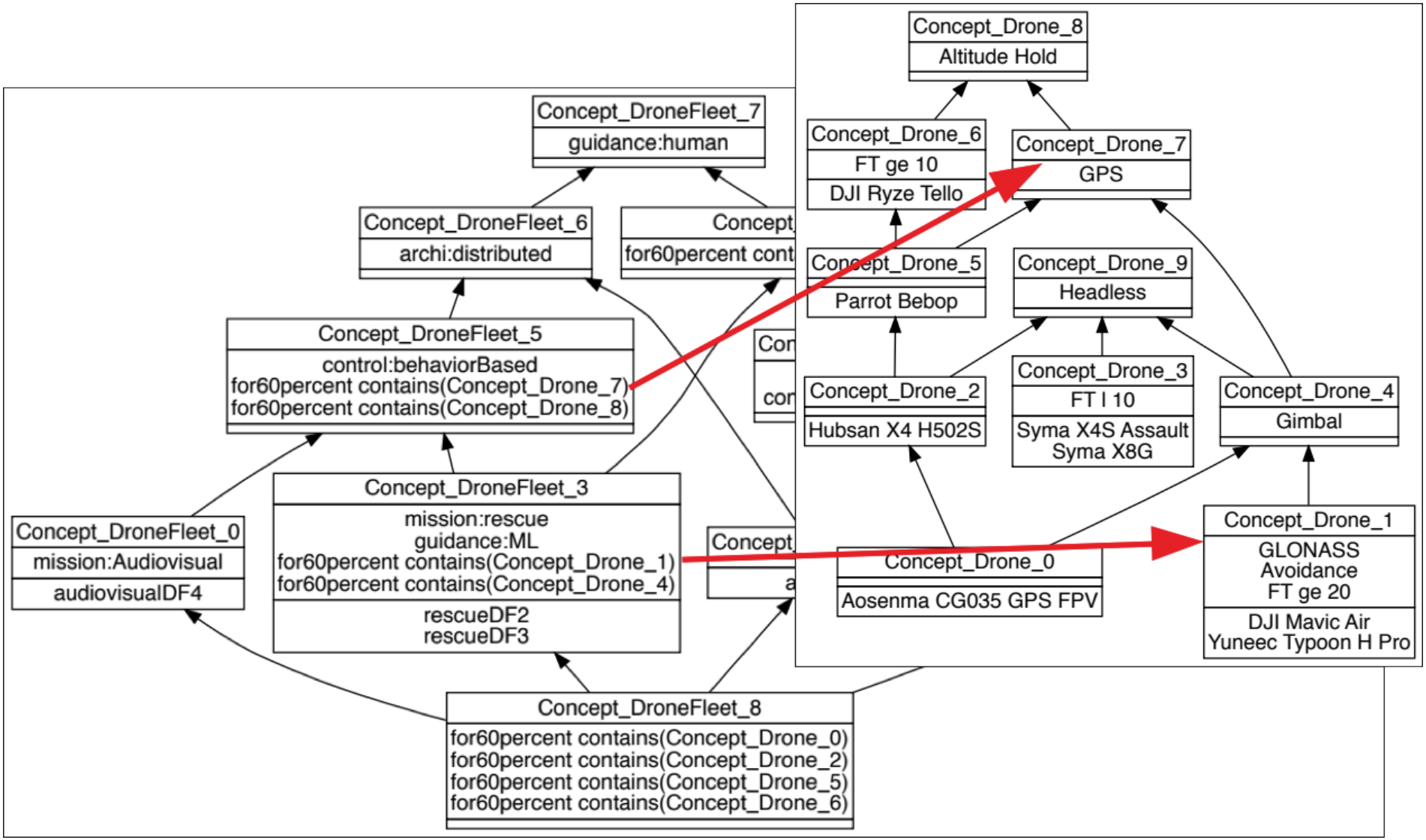


Drone Fleet	Examples of Relational Attributes
agriDF1, rescueDF2	$\exists$ contains (Concept_Drone_1)
rescueDF2, rescueDF3	$\exists\forall$ contains (Concept_Drone_1)
agriDF1	$\exists\forall\geq 50\%$ contains (Concept_Drone_1)
aud.vis.DF4	$\exists\forall\geq 40\%$ contains (Concept_Drone_1)
agriDF1, aud.vis.DF4	$\exists\supseteq$ contains (Concept_Drone_1)
rescueDF2, rescueDF3	$\exists\supseteq\geq 50\%$ contains (Concept_Drone_1)

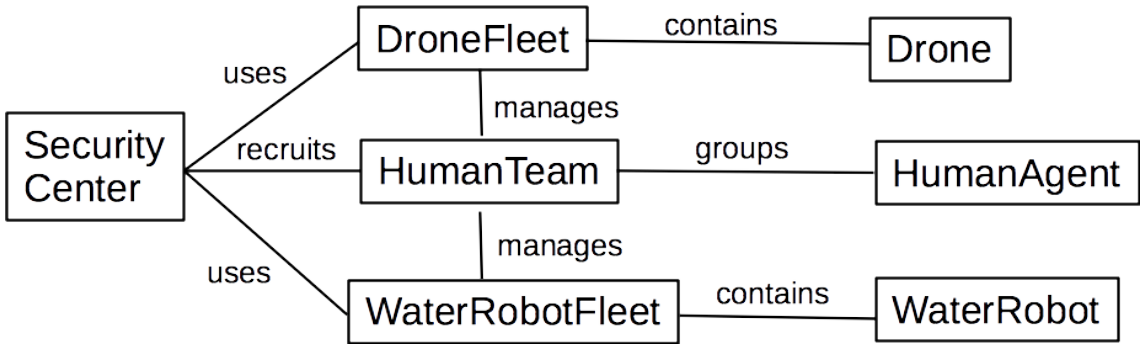
Les flottes de drones étendues par les relations vers les drones

DroneFleet	mission:agri	mission:rescue	mission:Audiovisual	archi:centralized	archi:distributed	guidance:human	guidance:ML	control:leaderFollower	control:behaviorBased	∃ ≥60% contains(Concept_Drone_0)	∃ ≥60% contains(Concept_Drone_1)	∃ ≥60% contains(Concept_Drone_3)	∃ ≥60% contains(Concept_Drone_2)	∃ ≥60% contains(Concept_Drone_4)	∃ ≥60% contains(Concept_Drone_5)	∃ ≥60% contains(Concept_Drone_6)	∃ ≥60% contains(Concept_Drone_7)	∃ ≥60% contains(Concept_Drone_9)	∃ ≥60% contains(Concept_Drone_8)
agriDF0	×			×		×		×				×						×	
agriDF1	×				×	×		×										×	
rescueDF2		×			×	×	×		×		×			×			×	×	×
rescueDF3		×			×	×	×		×		×			×			×	×	×
audiovisualDF4			×		×	×			×								×		×

Les flottes de drones étendues par les relations vers les drones



Cas général



Le processus peut itérer

- Modèles complexes avec des chemins ou circuits de toute taille
- Les concepts sont propagés le long des chemins et des circuits étape après étape
- Le processus s'arrête quand aucun nouveau concept n'est découvert

Tool

- http://dataqual.engees.unistra.fr/logiciels/rcaExplore

Relational Concept Analysis - Application à UML

Modèle ER

- Catégories/entités (formal contexts ou Object-Attribute contexts): classes, attributes, operations, roles
- Relations (relational contexts ou Object-Object contexts): hasAttribute, hasRole, hasOperation, hasTypeEnd

Reified characteristics: Object-Attribute contexts

Kclass
RainGauge
Anemometer
Rainfall
Wind
RainReport

Krole	measuredRainfall	measuredWind	storedRainfall
RG::measuredRainfall	×		
RR::storedRainfall			×
A::measuredWind		×	

Koperation	print	MC	MCWA	MCWSWD	PstoredRF
RR::print	×				×
R::printinfo	×	×	×		
W::printinfo	×	×		×	

Kattribute	tubeHeight	measureInterval	precision	measuringDate	codeQuality	waterAmount	windStrength	windDirection	period	number	int	real
RG::tubeHeight	×											
A::measureInterval		×										
A::precision			×									
R::measuringDate				×								
W::measuringDate				×								
R::codeQuality					×					×	×	
W::codeQuality					×					×		×
R::waterAmount						×						
W::windStrength							×					
W::windDirection								×				
RR::period									×			

Reified characteristics: Object-Object contexts

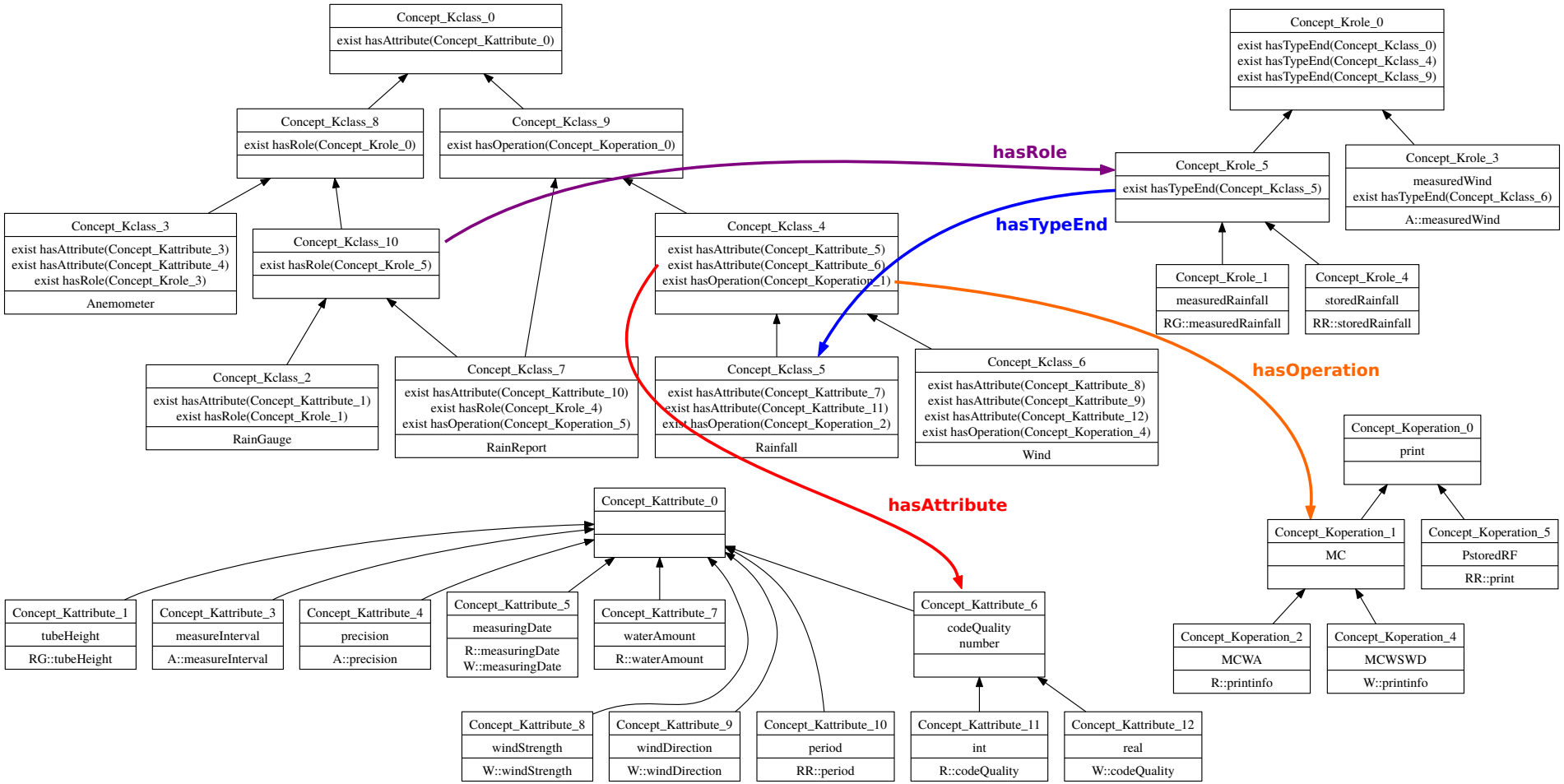
hasAttribute	RG::tubeHeight	A::measureInterval	A::precision	R::measuringDate	W::measuringDate	R::codeQuality	W::codeQuality	R::waterAmount	W::windStrength	W::windDirection	RR::period
RainGauge	×										
Anemometer		×	×								
Rainfall				×		×		×			
Wind					×		×		×	×	
RainReport											×

hasRole	RG::measuredRainfall	RR::storedRainfall	A::measuredWind
RainGauge	×		
Anemometer			×
Rainfall			
Wind			
RainReport		×	

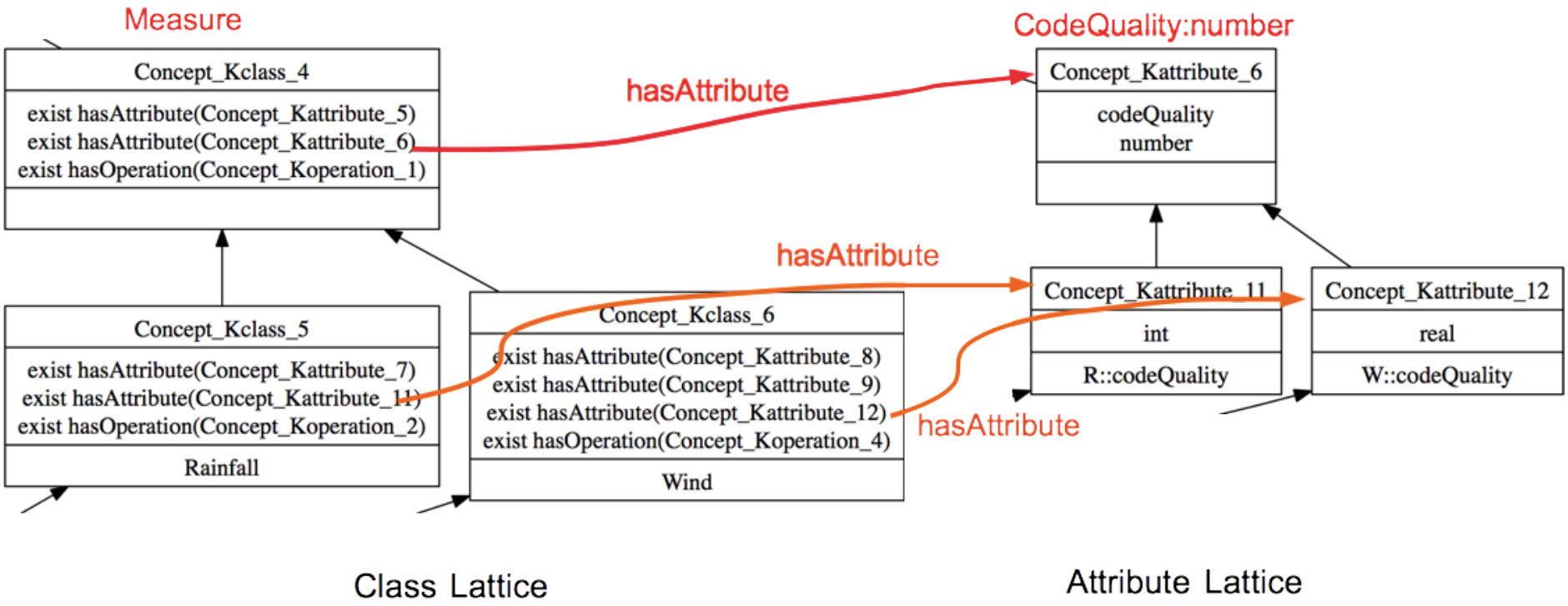
hasOperation	RR::print	R::printinfo	W::printinfo
RainGauge			
Anemometer			
Rainfall		×	
Wind			×
RainReport	×		

hasTypeEnd	**Rainfall**	**Wind**
RG::measuredRainfall	×	
RR::storedRainfall	×	
A::measuredWind		×

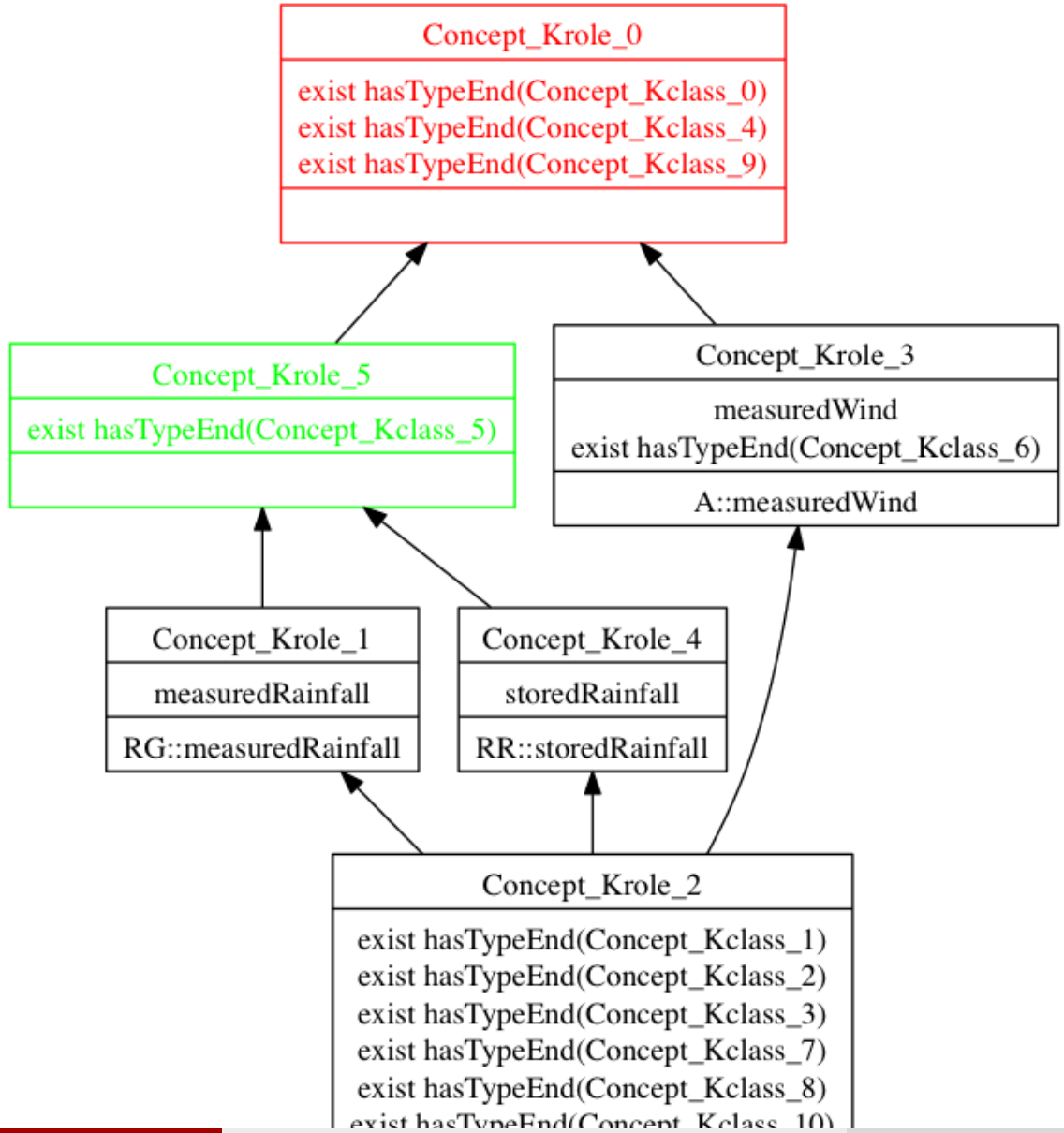
Reified characteristics: treillis



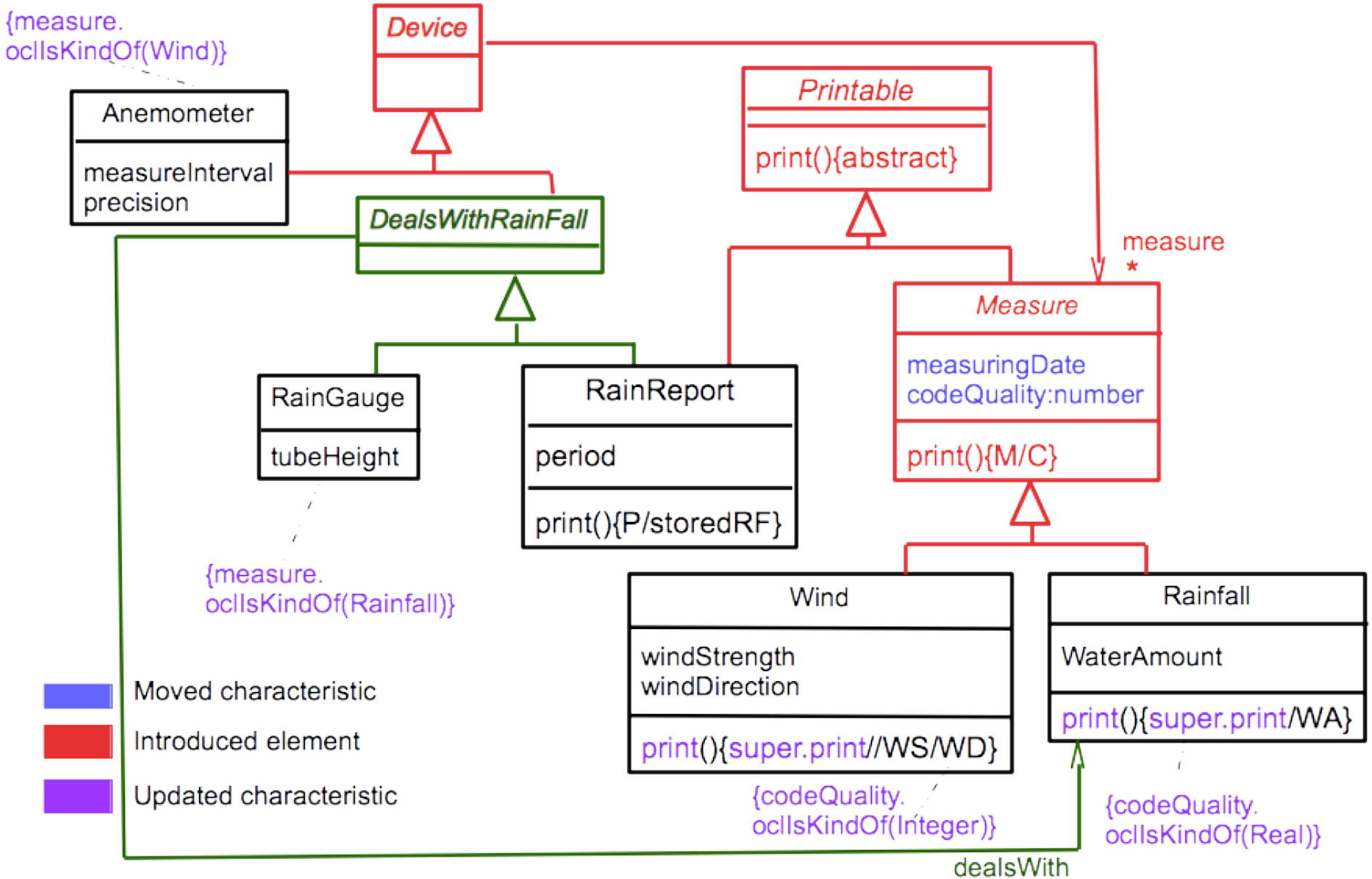
Reified characteristics: détail sur classes et attributs



Reified characteristics: treillis des roles



Reified characteristics: modèle revisité



Clustered reified characteristics

Krole1	measuredRainfall	measuredWind
RG::measuredRainfall	×	
A::measuredWind		×

Krole2	storedRainfall
RR::storedRainfall	×

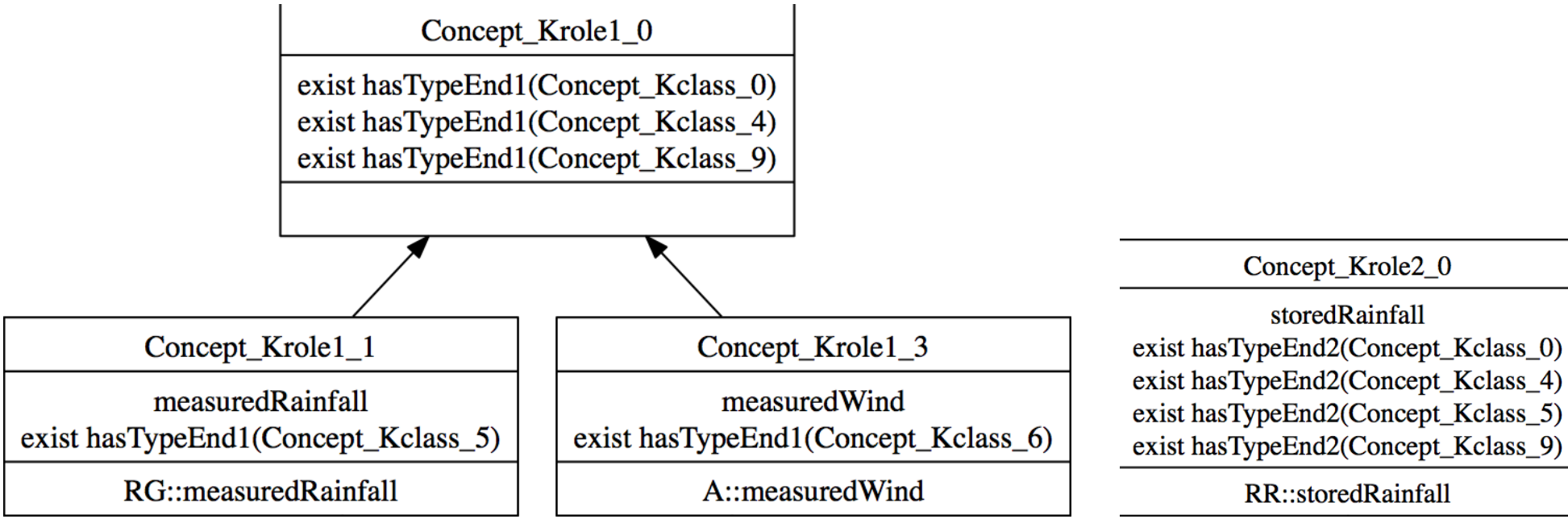
hasTypeEnd1	Rainfall	Wind
RG::measuredRainfall	×	
A::measuredWind		×

hasTypeEnd2	Rainfall	Wind
RR::storedRainfall	×	

hasRole1	RG::measuredRainfall	A::measuredWind
RainGauge	×	
Anemometer		×
Rainfall		
Wind		
RainReport		

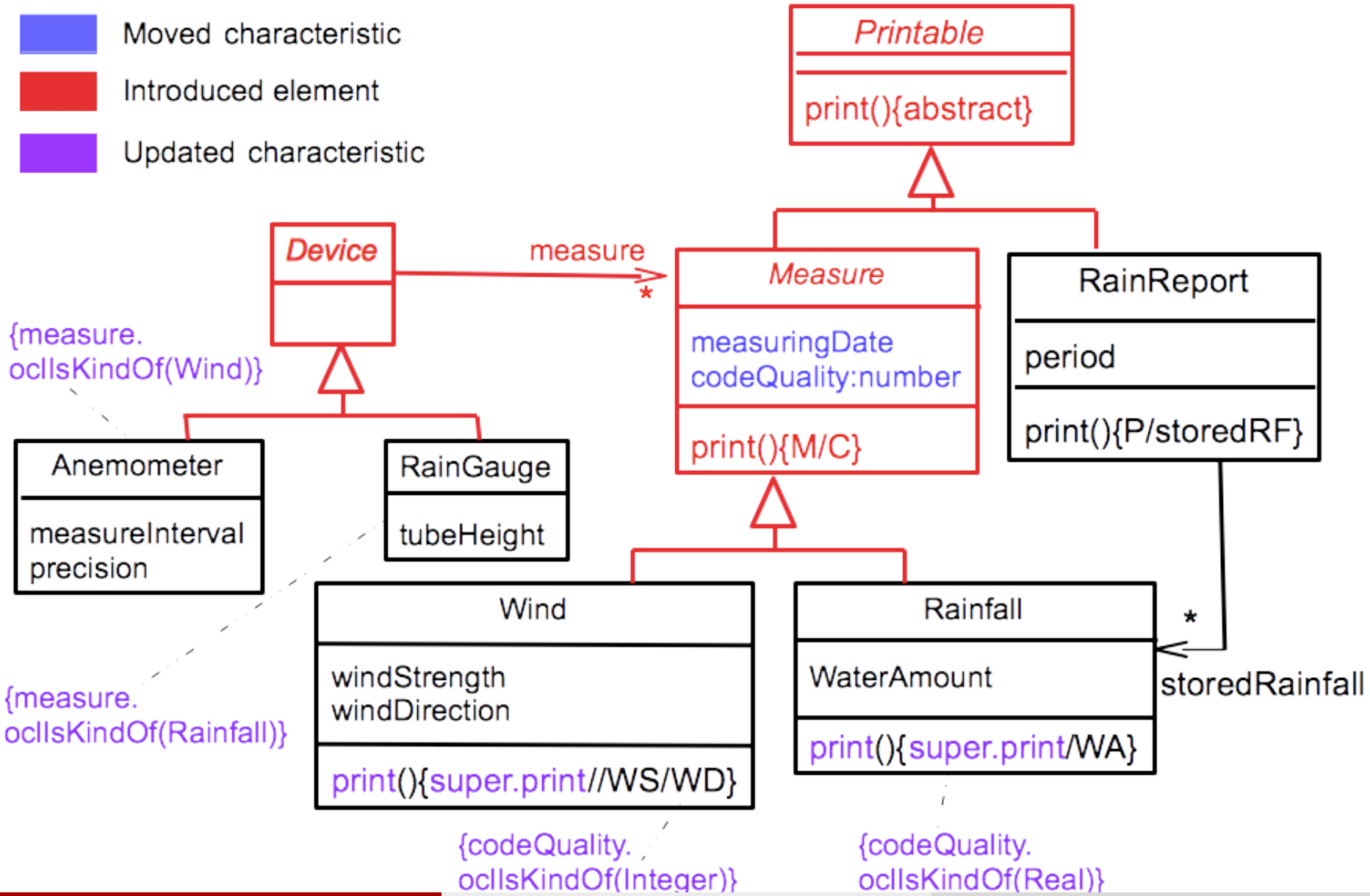
hasRole2	RR::storedRainfall
RainGauge	
Anemometer	
Rainfall	
Wind	
RainReport	×

Clustered characteristics: Séparation des rôles



La même approche doit être appliquée aux attributs et opérations pour éviter la sur-généralisation (ex. éviter d'introduire *DealsWithRainFall*)

Clustered characteristics: Forme normale



Synthèse

- FCA pour factoriser des descriptions sans relations
  - En UML, prise en compte de caractéristiques, y compris taxonomiques
- RCA pour factoriser des descriptions incluant des relations
  - En UML, prise en compte des associations du méta-modèle