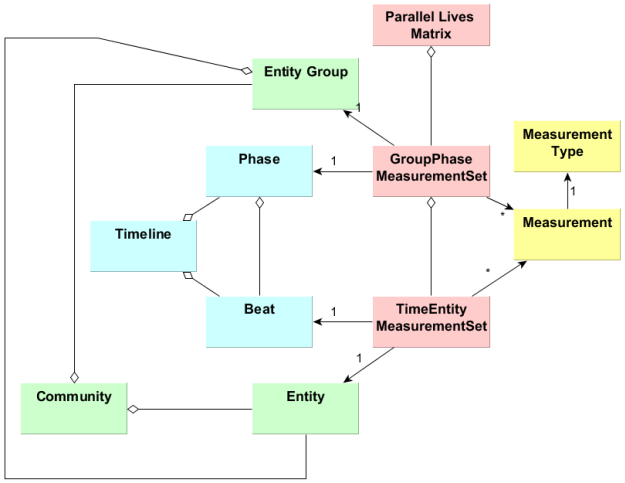
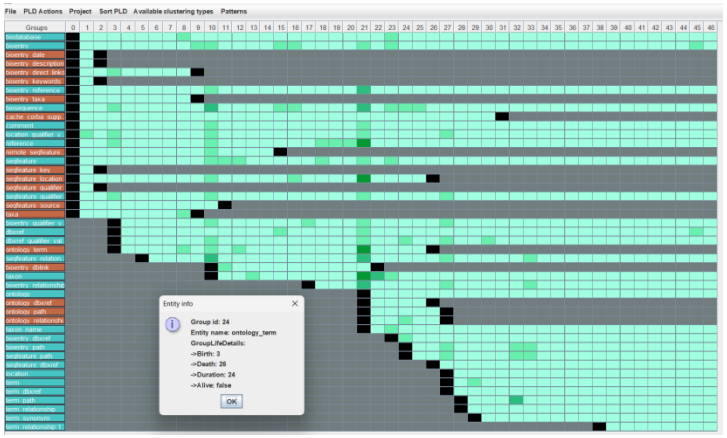




Parallel Lives Diagrams for Co-Evolving Communities and their Application to Schema Evolution

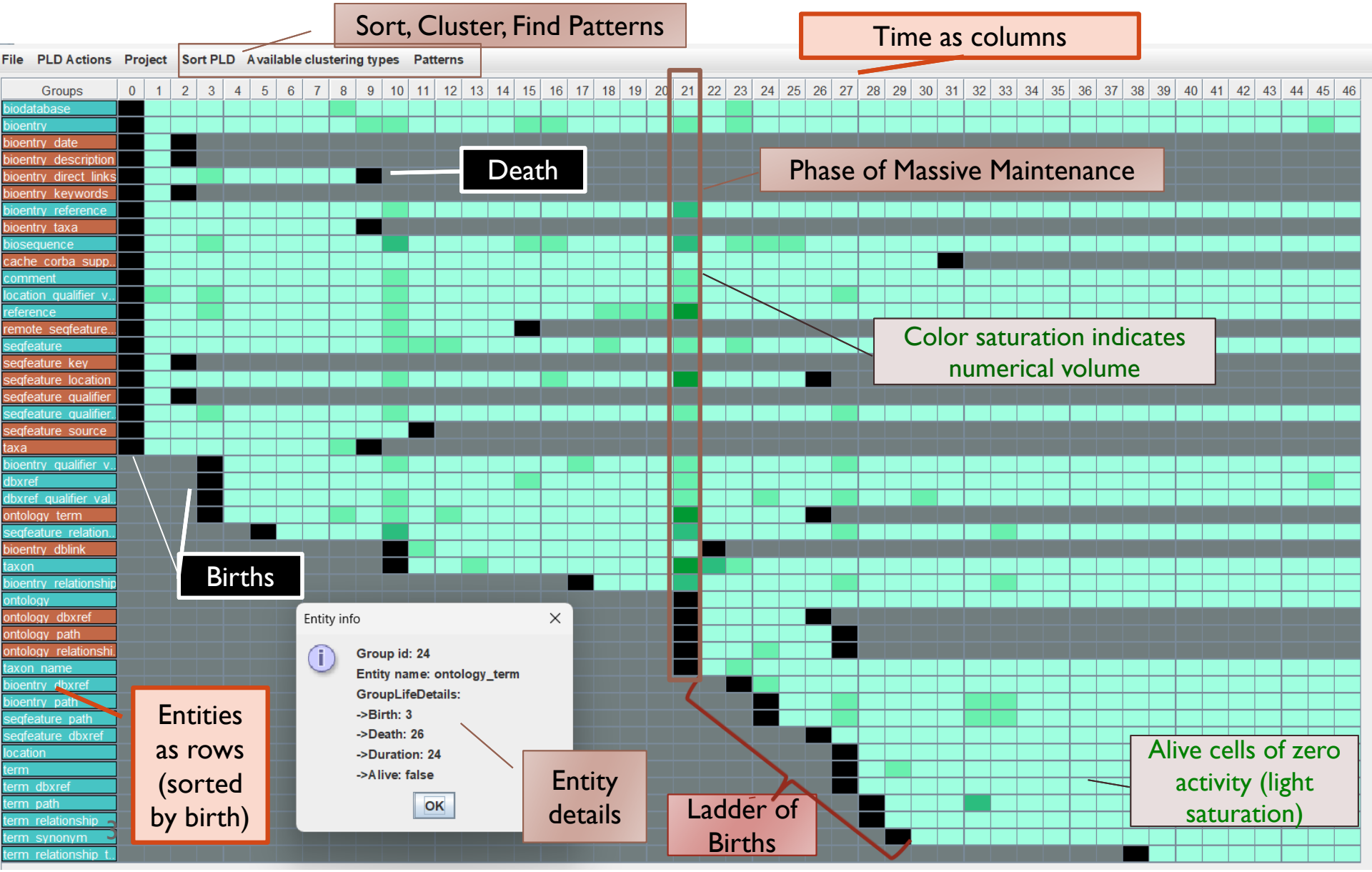
Fanis Giachos, Nikos Pantelidis, Christos Batsilas, Apostolos Zarras, and Panos Vassiliadis

http://www.cs.uoi.gr/~pvassil/projects/schemaBiographies/

Dep. of Computer Science & Engineering
University of Ioannina - Greece
www.cs.uoi.gr

Context and problem formulation

- How do we represent, and visualize, the way different entities of a community evolve together over time?
  - How do tables of a relational schema evolve together over time?
  - How do classes of an object-oriented program evolve together over time?
  - How do functions and data structures of a C program evolve together over time?
  - How do entities, controllers and views of Web application evolve together over time?
  - ….

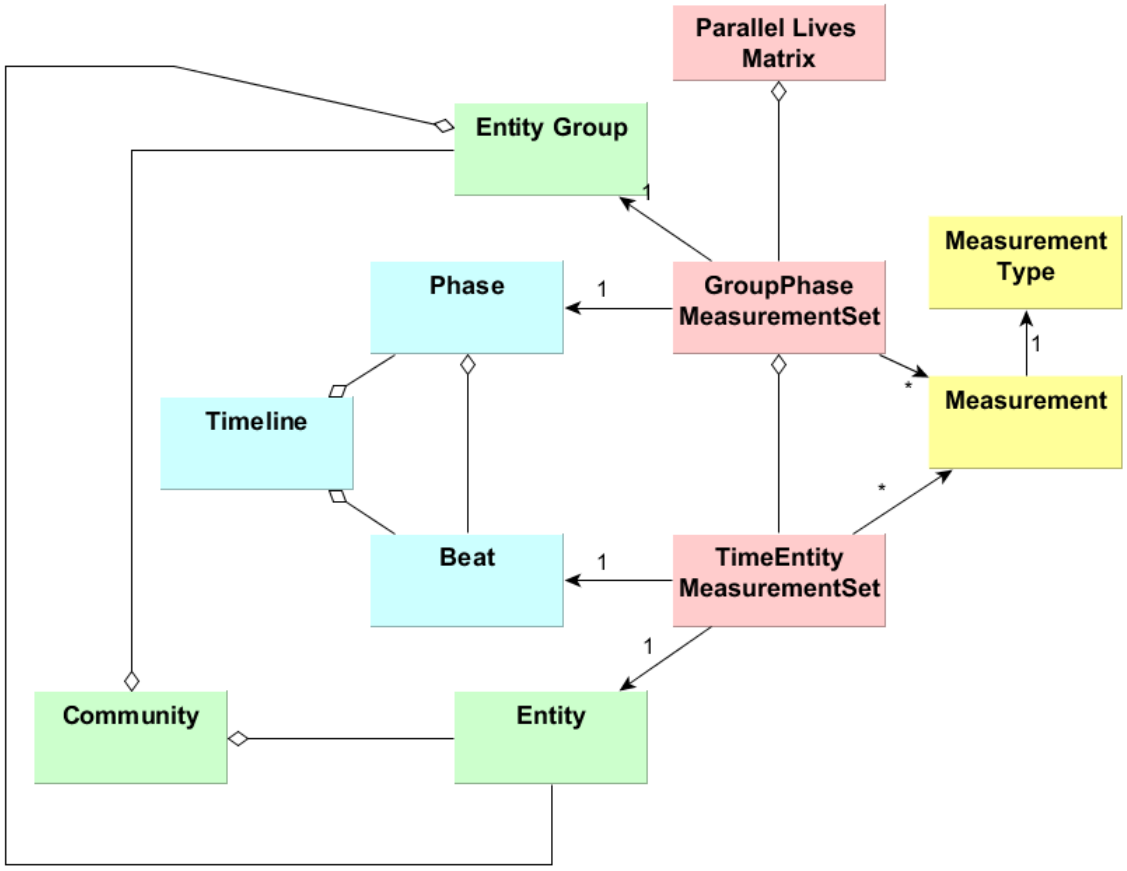

Sort, Cluster, Find Patterns
Time as columns
File
PLD Actions
Project
Sort PLD
Available clustering types
Patterns
Groups
Death
Phase of Massive Maintenance
Color saturation indicates numerical volume
Births
Entity info
Group id: 24
Entity name: ontology_term
GroupLifeDetails:
->Birth: 3
->Death: 26
->Duration: 24
->Alive: false
OK
Entities as rows (sorted by birth)
Entity details
Ladder of Births
Alive cells of zero activity (light saturation)

Contribution

- A **conceptual model** that represents the lives of co-evolving entities along:
  - A **timeline**, and a set of peer **entities**, forming **a two dimensional space** and **context** for the common life of a community, and,
  - **quantifiable measures** of activity for each of them.
- A **tool** (Plutarch's Parallel Lives) that is based on the **model** and allows a quick **understanding** (and reporting) of how the life of the **community has evolved**.
- The facilitation of the fully automated **identification of highlights, or patterns**, over the PLD.

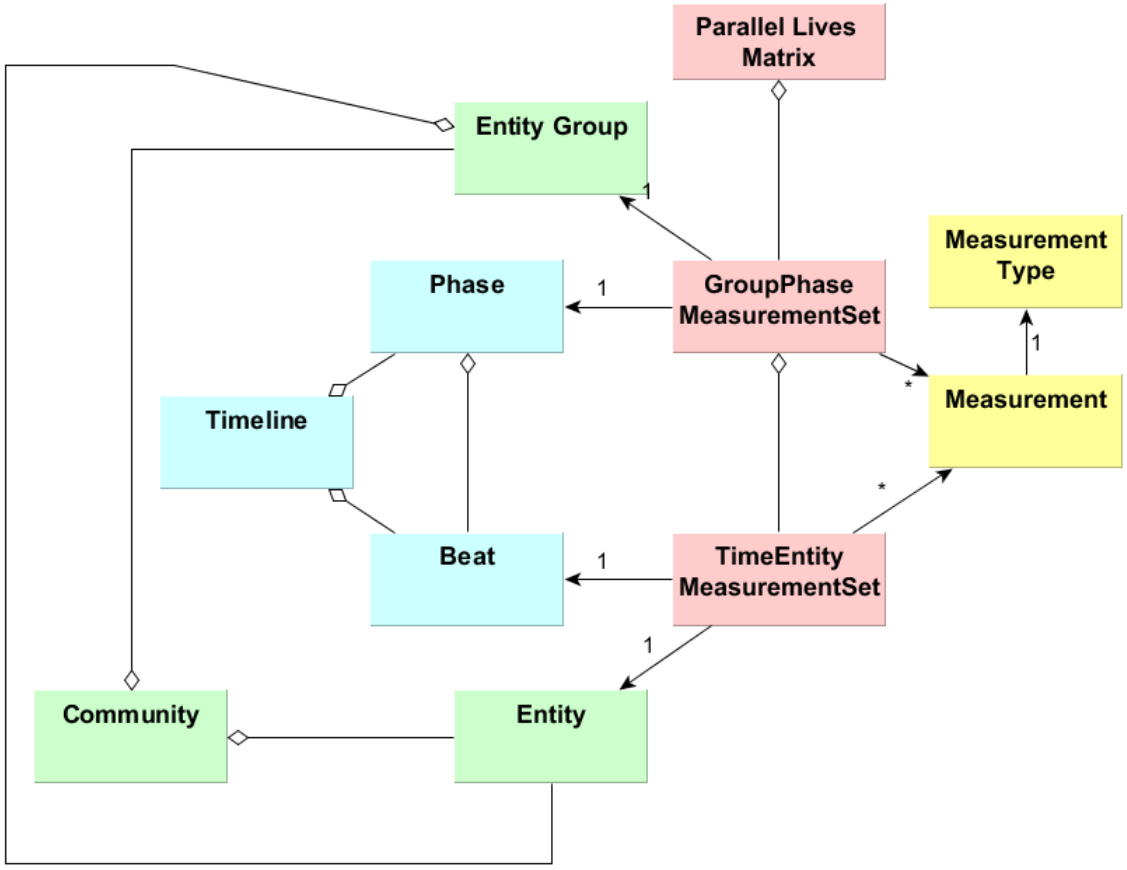
The model

Time Concepts



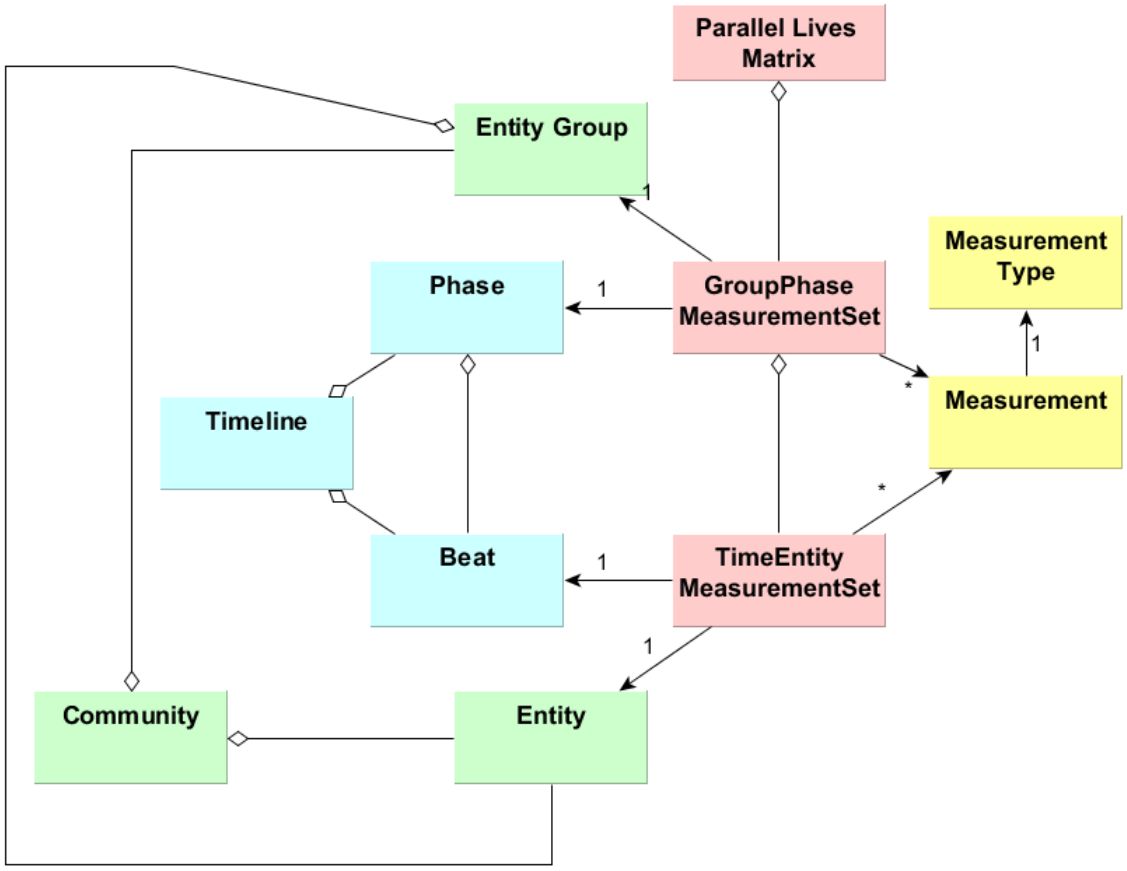
- **Beat**: individual time point granule
  - Seconds, milliseconds, commits, etc.
- **Timeline**: sequence of beats.
- **Phase**: period of time points, to allow time zoom out.
  - Roll-up of time to e.g., days or months, years, etc.

Community Concepts



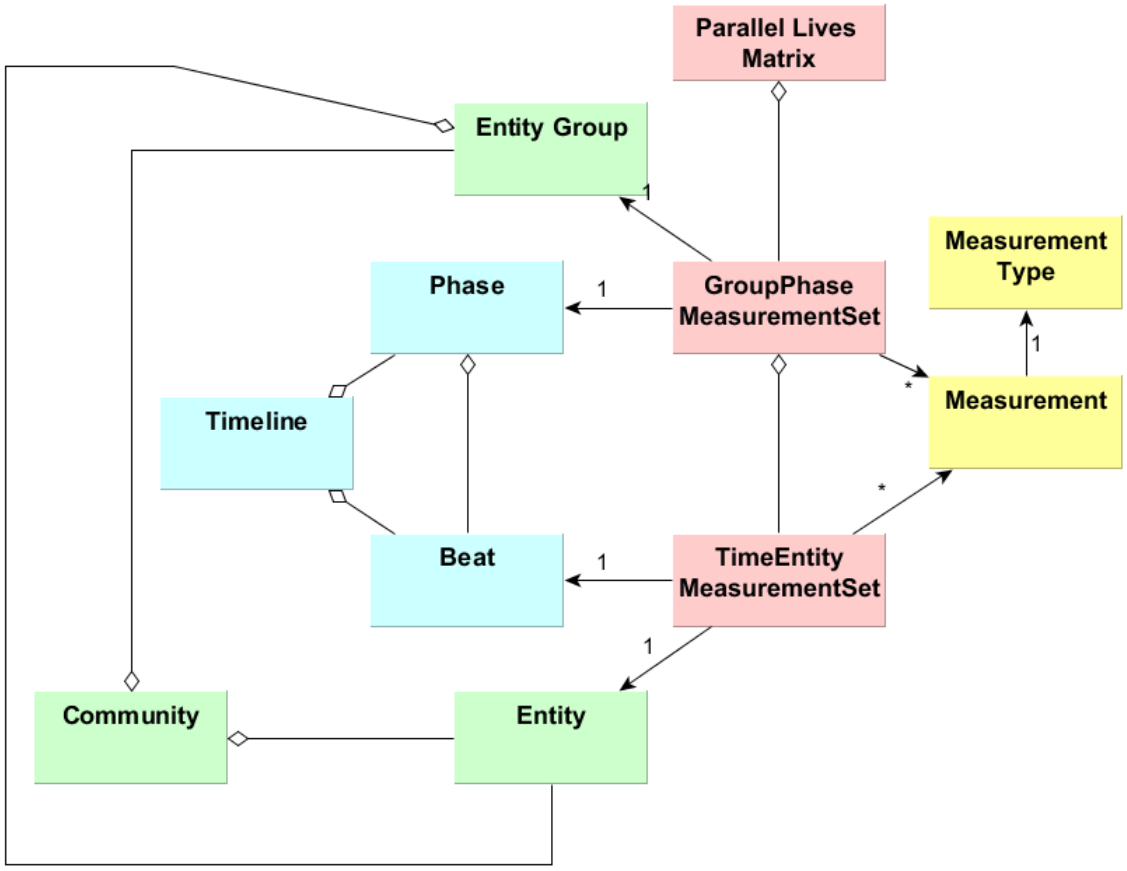
- **Entity**: individual, granular, elements.
  - A relation, a class, a function, etc.
- **Community**: set of entities, evolving together.
  - A schema, a software project, etc.
- **Entity group**: group of related entities, to allow community zoom out.
  - A package, a module, etc.

Measurement Concepts



- **Measurement Type**: individual, granular, measurement of change
  - Number of attributes injected, deleted, updated over their data type, etc.
  - Number of lines ins/del/upd is a file.
  - Number of methods/attributes/… ins/del/upd in a class.
- **Measurement**: the number measured.

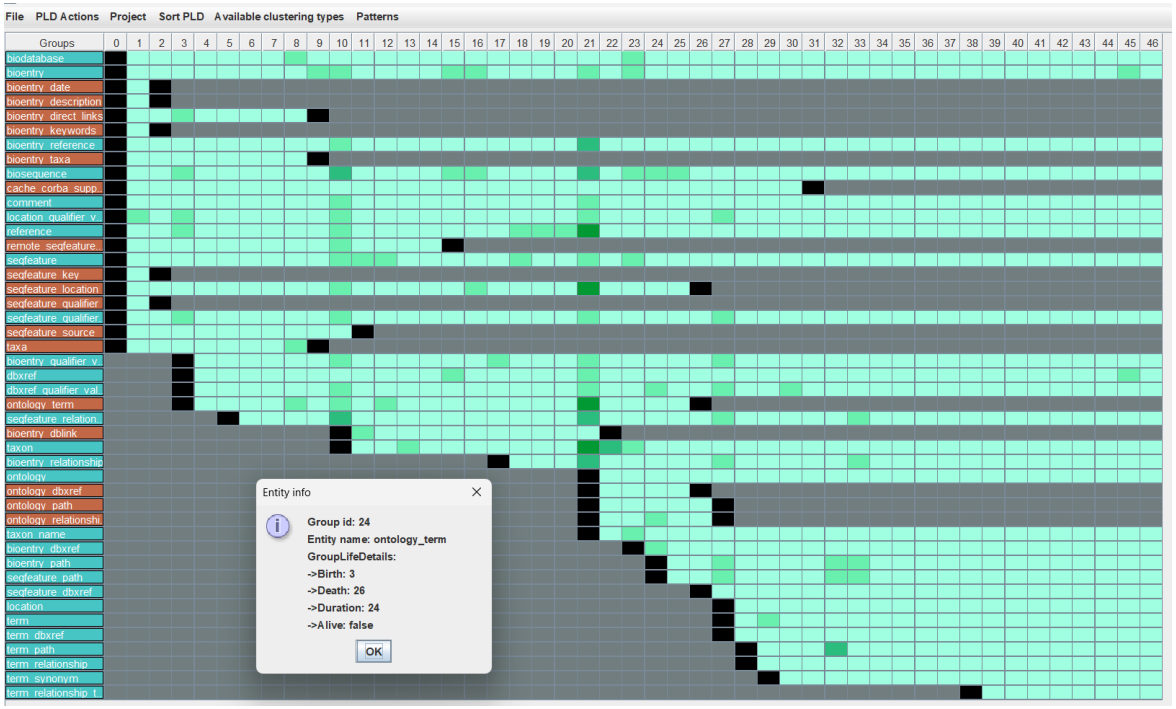
Relations between Concepts



- **TimeEntityMeasurement (TEM)**: relates a measurement of a certain type with a combination of entity and beat.
  - The total number of attributes changed for table Employee in 1st June 2023 was 0. (assuming day to be the time granule)
- **GroupPhaseMeasurment**: aggregate measurement for a zoomed-out context
  - The total number of attributes changed for table Employee in June 2023 was 2. (roll-up time to month)

Parallel Lives Diagrams

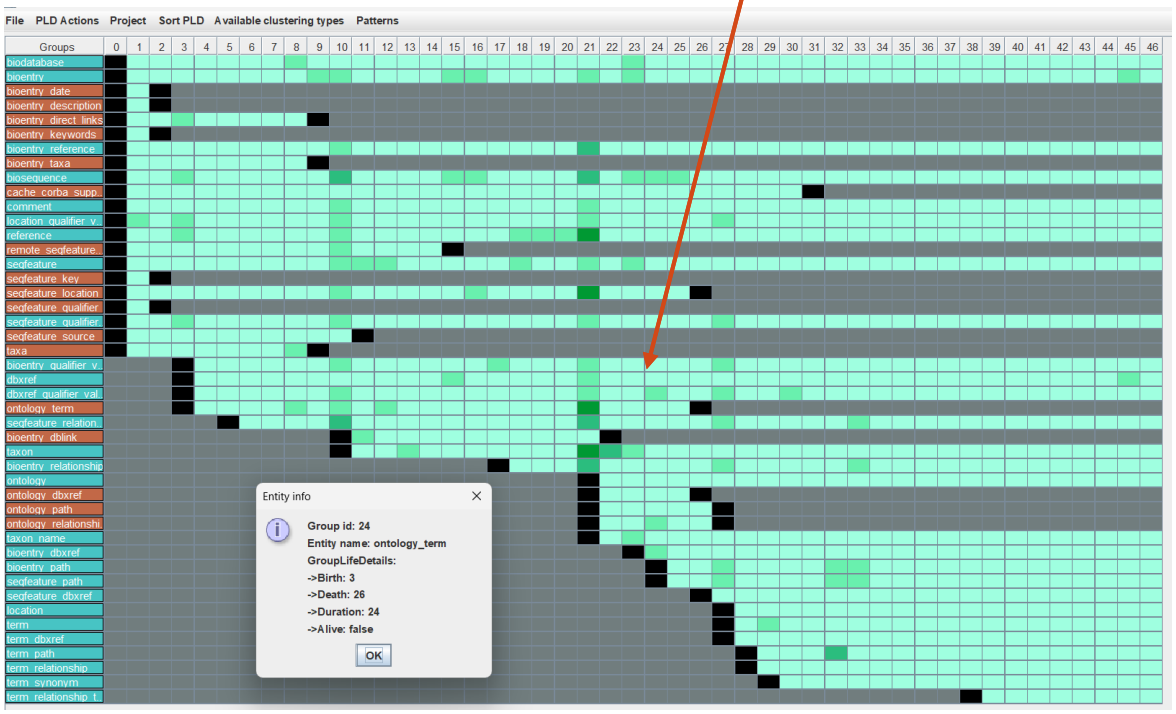
Time * Community create a 2D space. Each TEM is a point (aka "cell") in the 2D space. We visualize it as a Parallel Lives Diagram



We can tell the **story** of a **community** with a simple, intuitive, single image !!!

TEM's have states

Every TEM has a **state**



Birth: marks the first appearance of e in t he community.

Active: e is a member of the community and evolves overtime

Disappearance: e leaves community for some time.

Inactive: e is not a member of the community for some time.

Rebirth: e appears again in the community.

Death: e leaves the commu nity permanently.

The highlights

Highlights

- In a PLD we can have **"areas of interest"** and **"highlights (patterns)"** in the 2D space depending on what we are looking for.

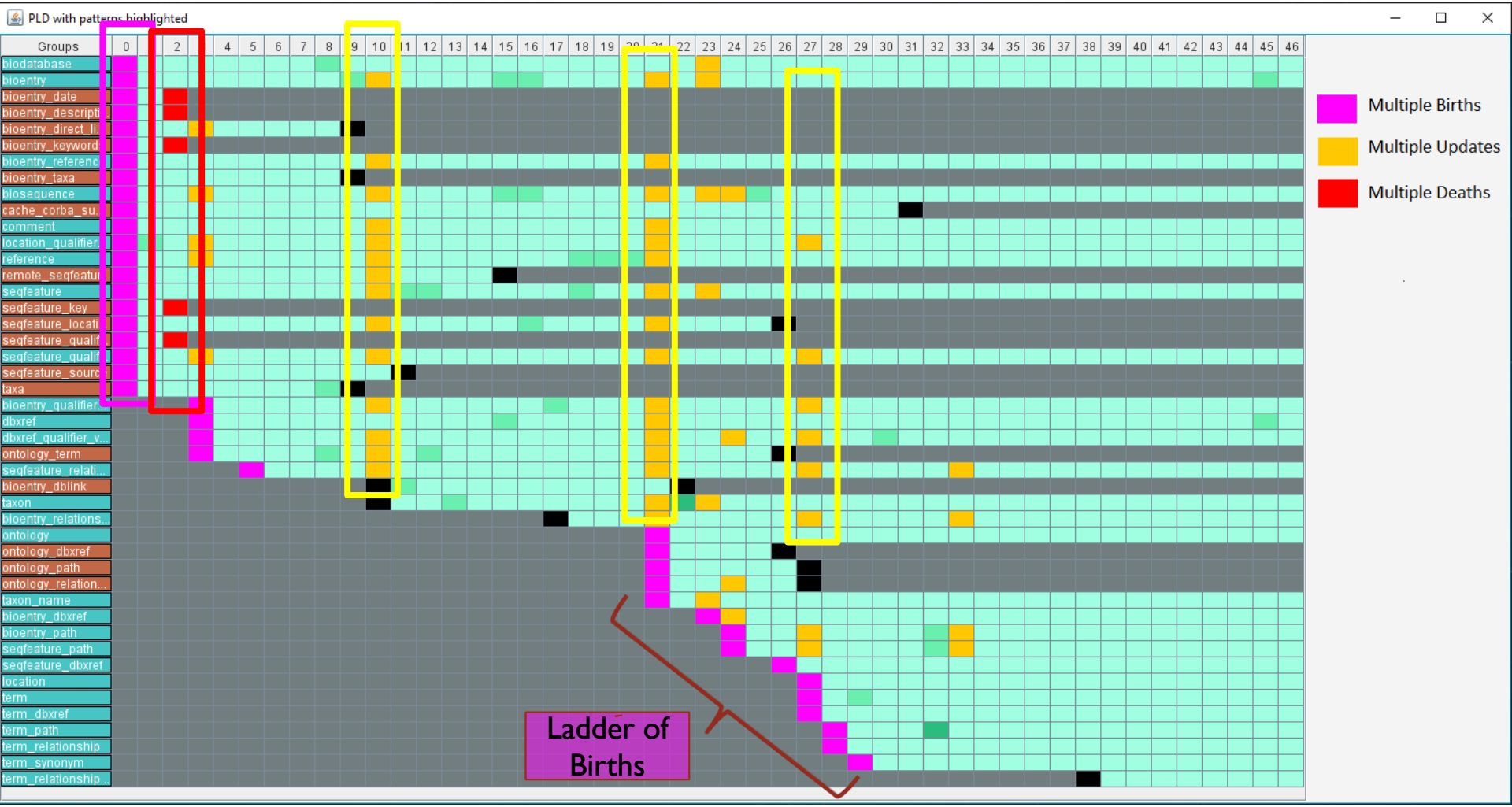
- We provide an **extensible set** of **patterns**, as well as **algorithms** to compute them.

Several Highlights

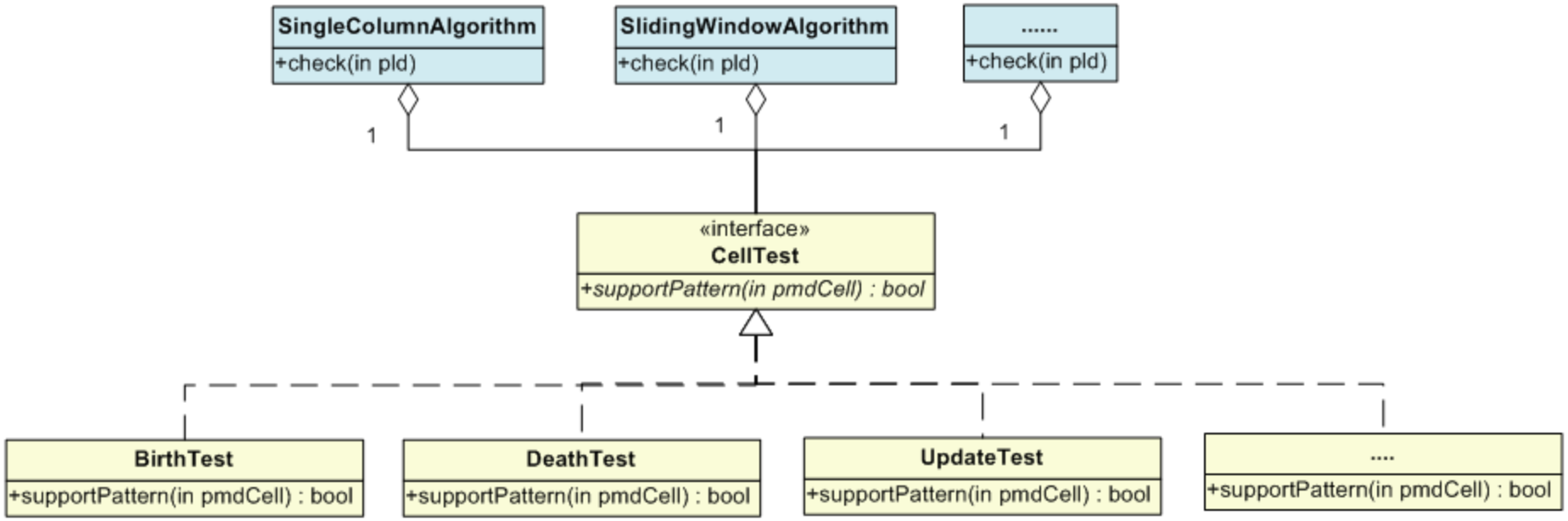
Massive Births

Massive Deaths

Massive Updates



Extensible set of Algorithms



A study over 195 FOSS schema histories

	Total # Columns	Total # Rows	# Columns In Patterns	# Rows In Patterns	%Col's in patterns	%Rows in patterns	#Total Patterns	#Births Patterns	#Deaths Patterns	#Upd. Patterns	#Stairs Patterns
MAX	516	283	36	253	1	1	33	10	4	16	3
MIN	2	1	0	0	0	0	0	0	0	0	0
SUM	2796	2872	410	2545	38.30	112.04	319	148	26	97	48
AVG	14.34	14.73	2.10	13.05	0.20	0.57	1.64	0.76	0.13	0.50	0.25
COUNT	195	195	195	195	195	195	195	195	195	195	195
Median	**4.00**	**6.00**	**1.00**	**5.00**	**0.12**	**0.85**	**1.00**	**1.00**	**0.00**	**0.00**	**0.00**
StdDevP	44.61	30.33	3.95	29.01	0.22	0.46	3.12	1.03	0.50	1.62	0.57
Mode	2	1	0	0	0	0	0	0	0	0	0
Count Non Zero	**195**	**195**	**120**	**120**	**120**	**120**	**120 (62%)**	**109 (56%)**	**18 (09%)**	**41 (21%)**	**38 (19%)**

- Patterns in 62% of the schemas,
  - Mostly massive birth patterns (56%) due to big-bang schema creations.
  - Deletions are rare.
  - Progressive expansion and massive maintenance is not very frequent.

Summary

Parallel Lives Diagrams

- A **conceptual model** that involves **entities**, **timelines**, **measurements** to capture how the different entities of a community co-evolve.
- The model allows:
  - The **visualization** and intuitive **understanding** of the community evolution in a simple, but also powerful, way.
  - the **mining** of **interesting highlights** of change that highlight important points and members in the evolution of the community.
- Future research:
  - Easier zoom-in/out with more flexible / predefined / … hierarchies
  - Pattern generalization – e.g., of birth staircases and massive updates to a "x changes soon after y" pattern is a simple example.
  - Automated reporting.

Thank you!

https://github.com/DAINTINESS-Group/

Everything is online!

Our group's git page

https://github.com/DAINTINESS-Group/

has links to **Data sets**

https://github.com/DAINTINESS-Group/Schema_Evolution_Datasets/tree/master/SchemaEvolutionDatasets2020

and Code

… for computing differences (Hecate)

… visualizing schema lives (Plutarch Par. Lives)

… visualizing the structure of FK's (Parmenidian Truth)

… handling the impact of evolution (Hecataeus)