GEFÖRDERT VOM

Bundesministerium
für Bildung
und Forschung

PXL 2.1: Toolkit for Physics Analyses in the Elementary Particle Physics

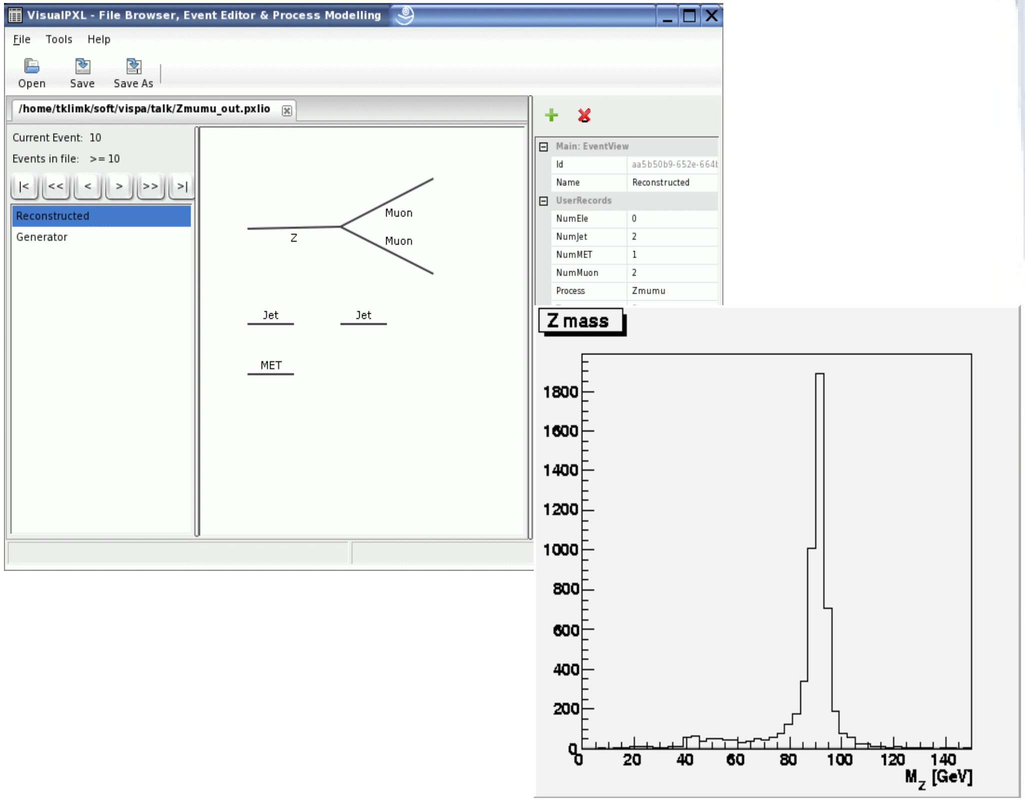
Tatsiana Klimkovich for the **VISPA** group

(O.Actis, M.Brodski, M.Erdmann, R.Fischer,
A.Hinzmann, T.Klimkovich, G.Müller, T.Münzer,
M.Plum, J.Steggemann, T. Winchen)

DPG Tagung, München, März 2009

Contents

- Physics Analysis in High Energy Physics experiment
- PXL: toolkit for physics analysis
- PXL key ingredients:
  - event container
  - relation management
  - user record
  - I/O
- Python interface



VisualPXL - File Browser, Event Editor & Process Modelling

/home/tklimk/soft/vispa/talk/Zmumu_out.pxlio

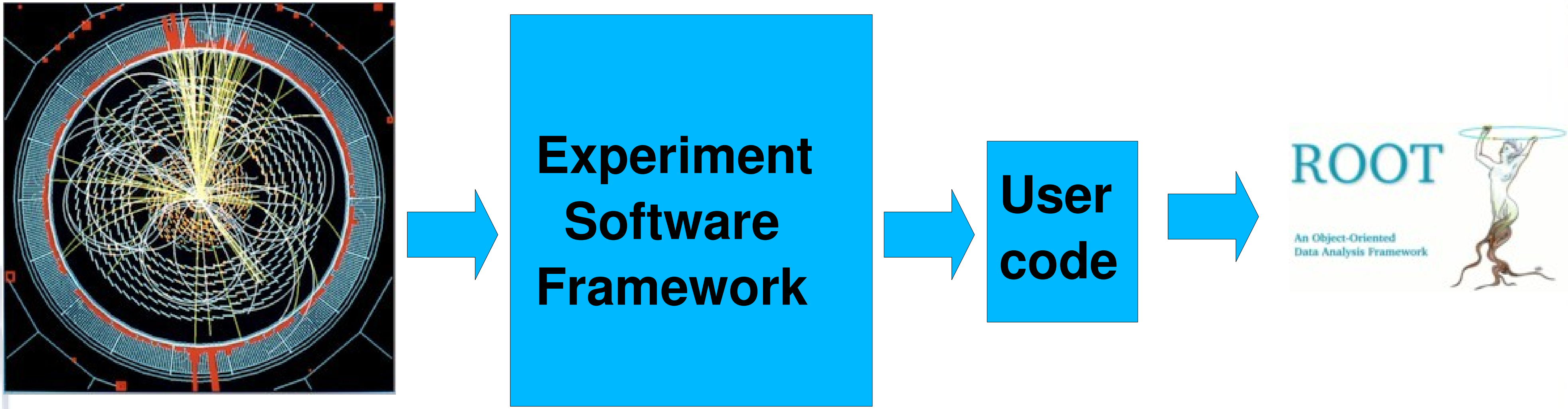
Current Event: 10

Events in file: >= 10

Main: EventView	
Id	aa5b50b9-652e-664t
Name	Reconstructed
UserRecords	
NumEle	0
NumJet	2
NumMET	1
NumMuon	2
Process	Zmumu

High Energy Physics Analysis

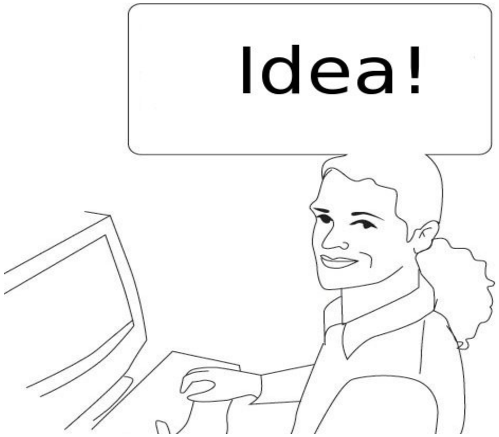
- During last years big achievements in developing analysis software for the experiments
- Experiments have different software frameworks e.g. H1OO in H1, CMSSW in CMS, ATHENA in ATLAS etc.
- On top of them more analysis specific software has been developed and used



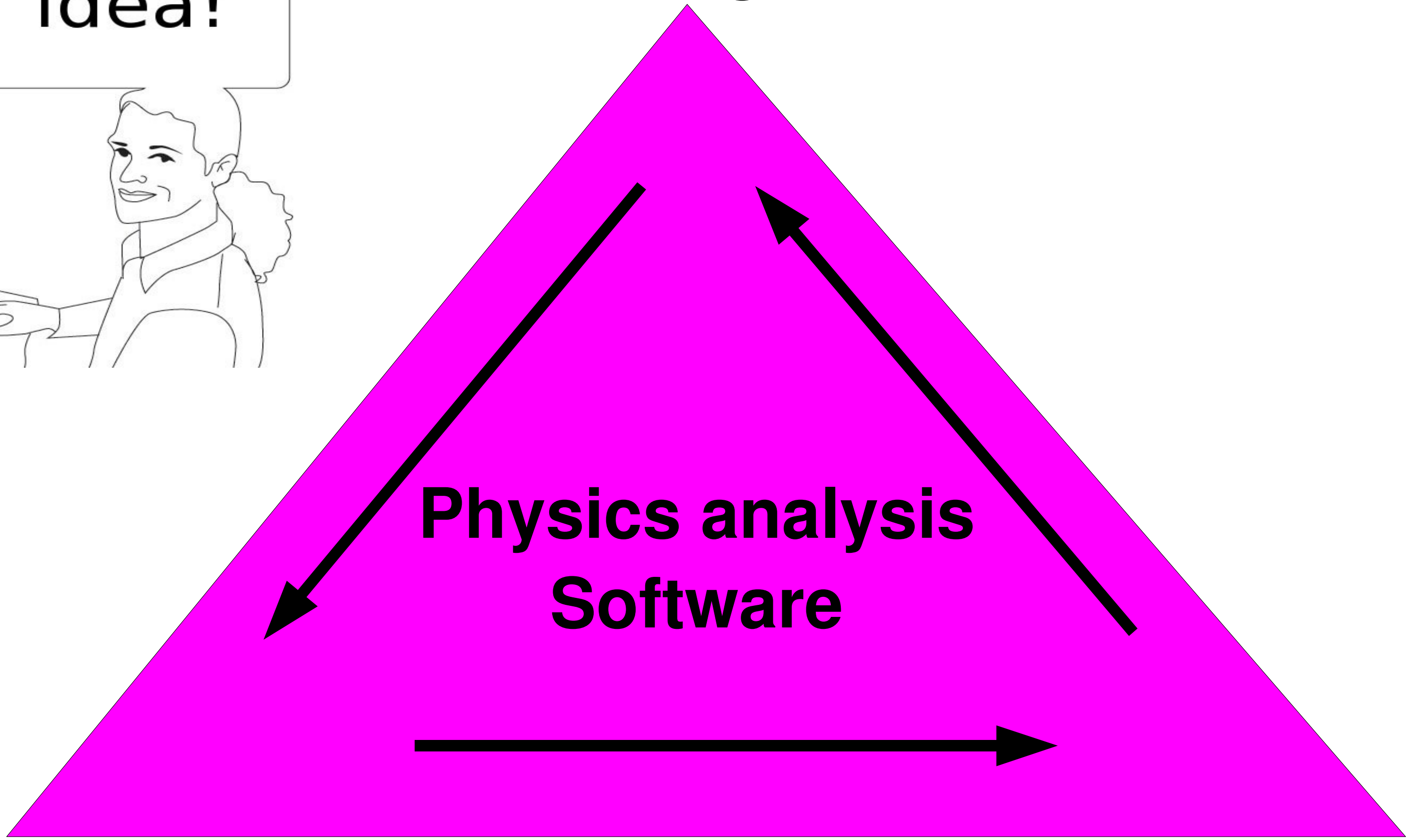
Experiment Software Framework

User code

Physics Analysis Flow



Prototyping
(design)



Physics analysis
Software

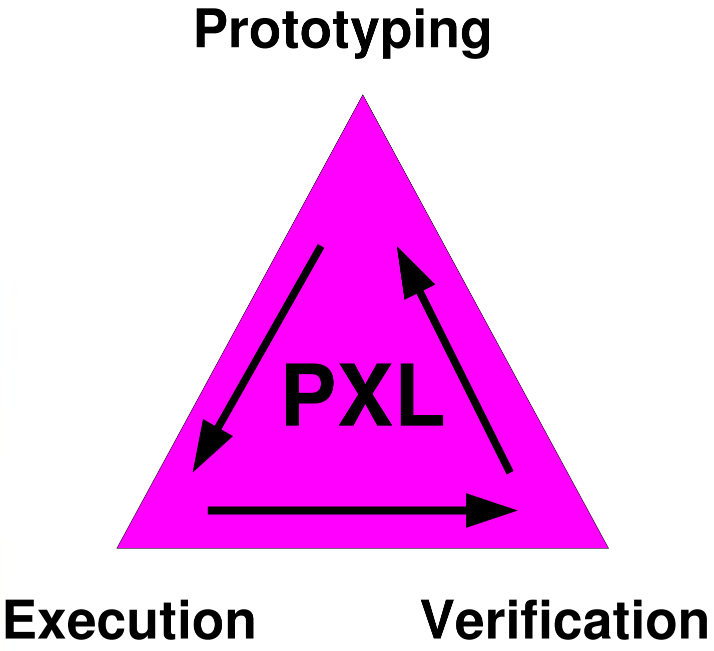
Execution
(steering)

Verification

Wish list of the analyser

- To have an easy way to develop analysis
- To start fast
- To dedicate minimal time for learning
- To have small summary data sets (ntuples)
- Possibility to perform analysis on the laptop, desktop and GRID
- To have fast I/O

High Energy Physics Analysis



Prototyping

PXL

Execution

Verification

VISPA

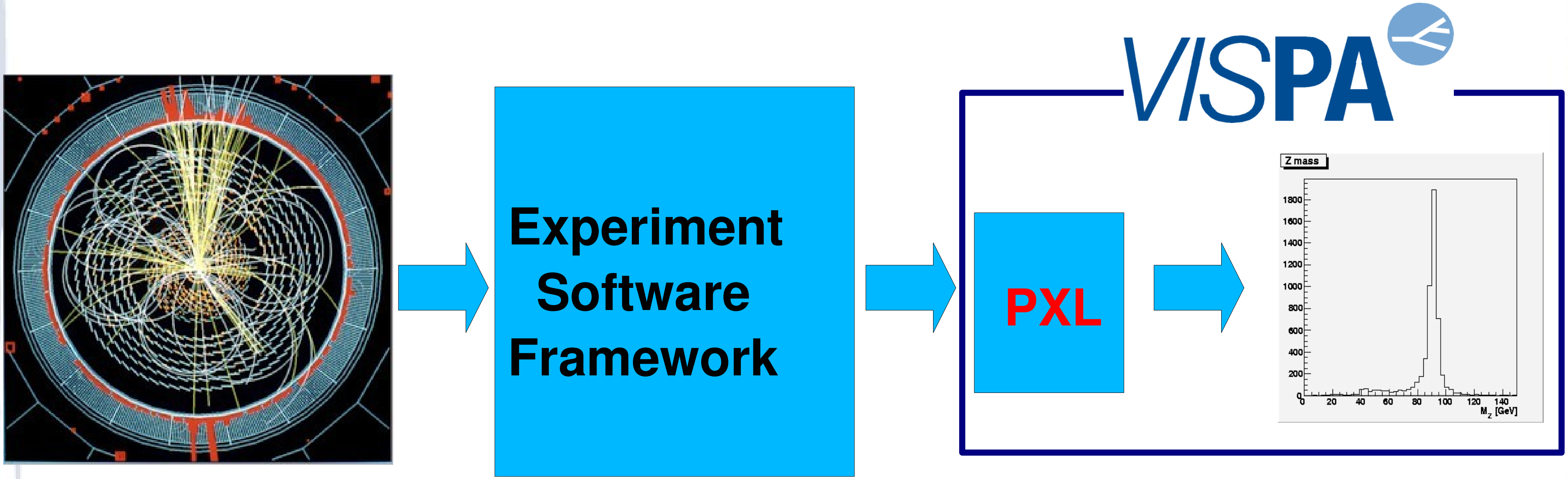
Visual Physics Analysis

Novel Concept of making physics analysis

(see next presentation of Andreas Hinzmann)

PXL (Physics eXtension Library)

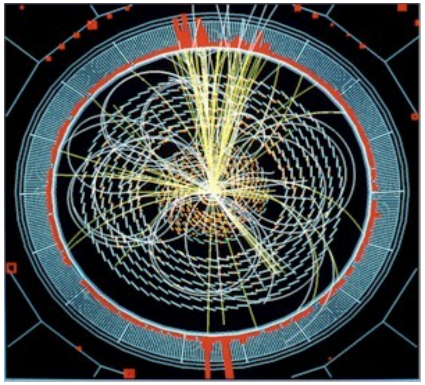
- C++ toolkit for high-level physics analysis
- Provides underlying physics analysis functionality for Visual Physics Analysis (VISPA)
- Version 2.1 (2009)
- Successor of PAX (Physics Analysis Expert) (2002-2007)

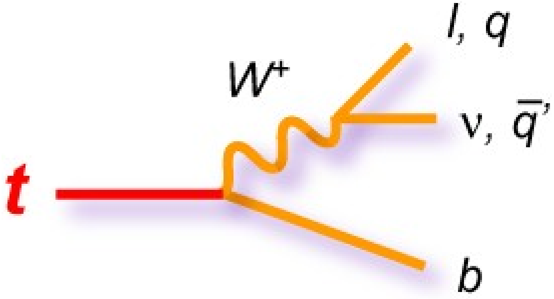


Experiment Software Framework

PXL

PXL key component: **Event Container**





- Particles (pxl::Particle)
- Vertices (pxl::Vertex)
- Collisions (pxl::Collision)
- User data (pxl::UserRecord)
- Their **relations** and **roles**

Physics objects

Event Interpretation

pxl::EventView

Event container pxl::Event can hold several **pxl::EventView**

Allows **deep copies** (physics objects with redirected relations, data members, user records)

PXL key component: **UserRecord**

- **All major PXL objects provide UserRecord for storage of user data** (pxl::UserRecord)**:**

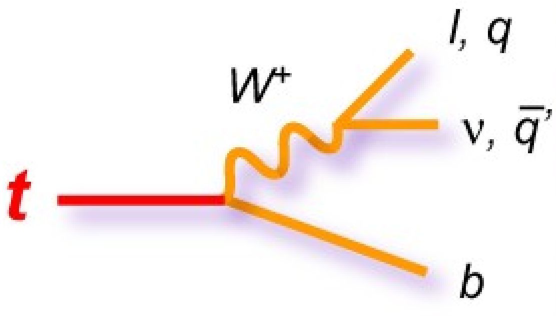
  - pairs of names and all basic C++ types (int, double, string, ...)

  - it can be e.g. data from the condition databases, btag information, true Monte Carlo information etc.

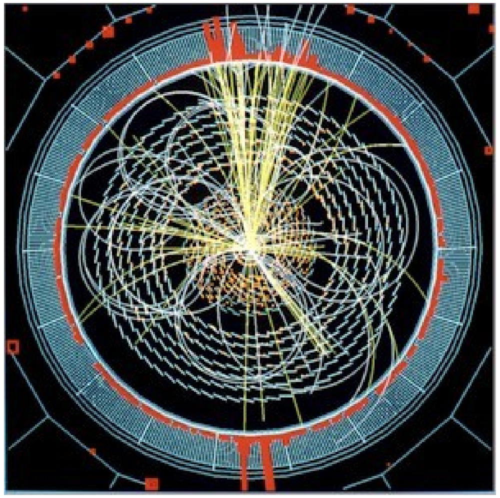
- **Deploys Copy-On-Write mechanism**
- **Flexible and simple extension of objects**

PXL key component: **Relation Management**

- **Mother, daughter and flat relations**
- **Safe removal in case of object deletion**



- **Possibility of relations e.g. between reconstructed and generated particles**



PXL key component: **Input / Output**

- **Main class pxl::Serializable**
- **Fast, Flexible**
- **Small file size: use ZLIB library for data compression**
- **Each object knows how to stream itself**
  - **methods “serialize” and “deserialize”**
- **Inclusion of user classes into I/O scheme**

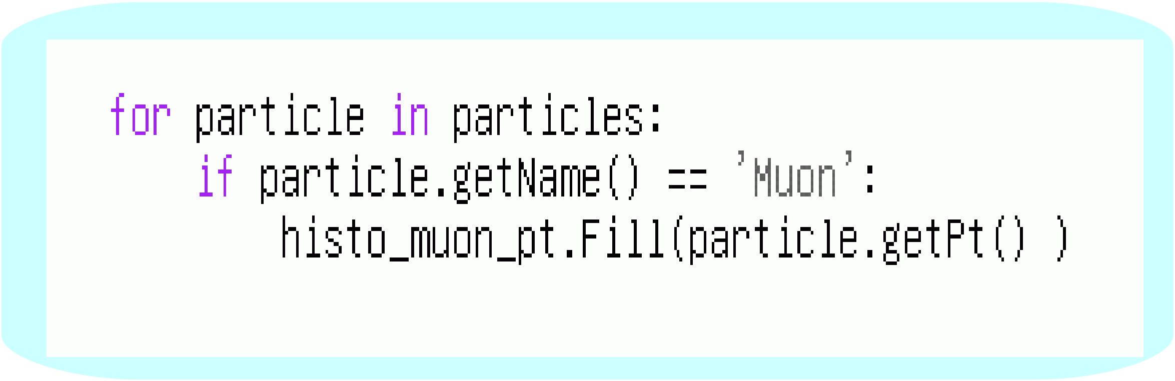
Python in HEP

- Python starts to be more popular for doing physics analyses
- Bender – LHCb Python-based physics analysis application
- Possible to perform analysis with Python in CMS
- Some use in D0

- Python code is easy to write and read
- Less code compared to C++
- Dynamic typing
- Automatic memory management
- Has an interactive mode for testing
- Object oriented, works on multiple platforms, open source

Python interface to PXL

- PyPXL: Python layer on PXL C++ for easy user syntax



```
for particle in particles:
    if particle.getName() == 'Muon':
        histo_muon_pt.Fill(particle.getPt() )
```

- Use of SWIG for automatic interface C++ → Python

Summary

- PXL is C++ toolkit for high-level physics analysis
- Key ingredients: event container, relation management, user record and fast I/O
- Can serve as an analysis software for any experimental analysis framework, e.g. VISPA
- Python interface to PXL
- Can work for any experiment
- First applications for CMS analysis, ILC starting

Summary II

- All software is continuously maintained
- Fully documented:

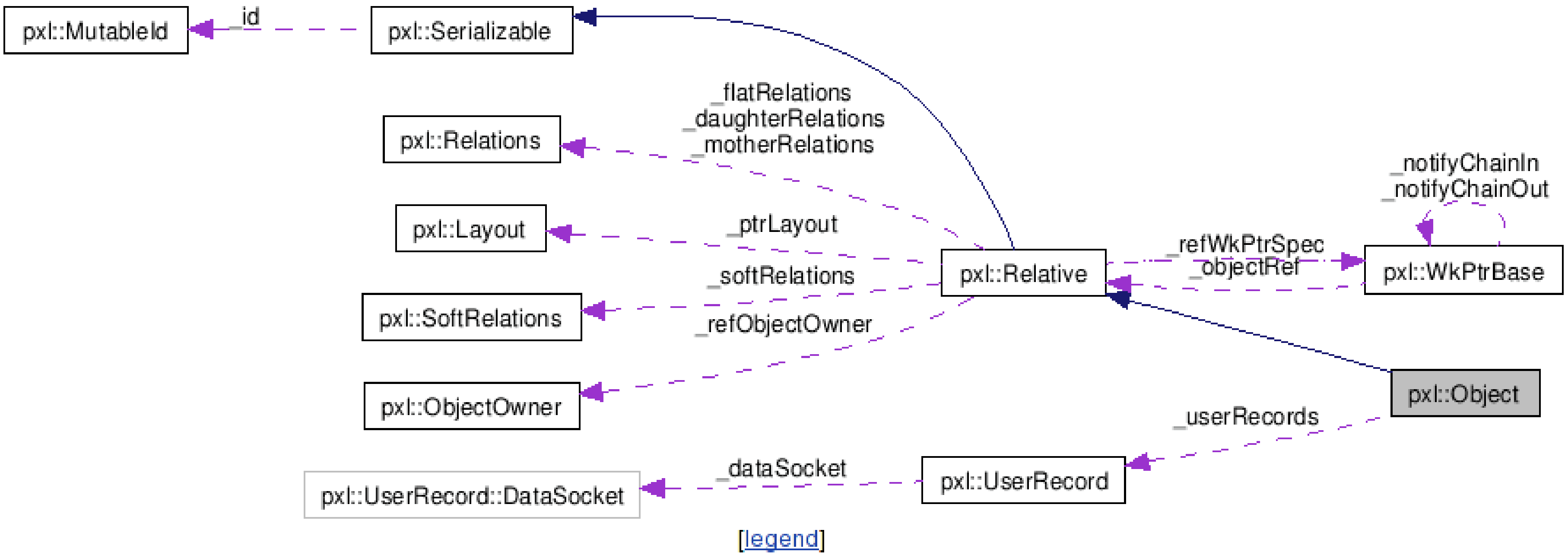
  - Manual for PXL:

    http://pxl.sourceforge.net/manual.pdf

  - Doxygen
- Available online at http://pxl.sourceforge.net
- Publications:

  http://arxiv.org/abs/0810.3609

PXL Objects Structure



- **Inheritance and composition**
  - **I/O** **(pxl::Serializable)**
  - **relations** **(pxl::Relative)**
  - **User data** **(pxl::Object)**
  - **Object container** **(pxl::ObjectManager)**