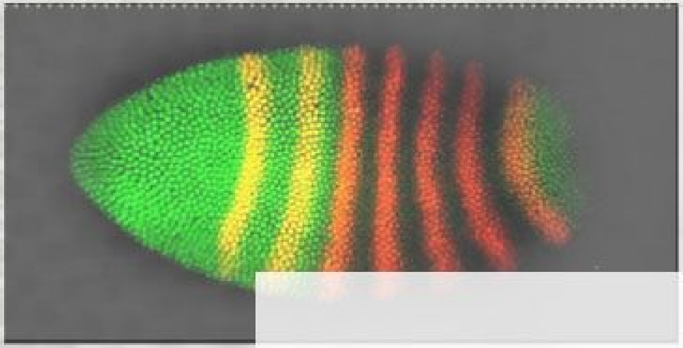
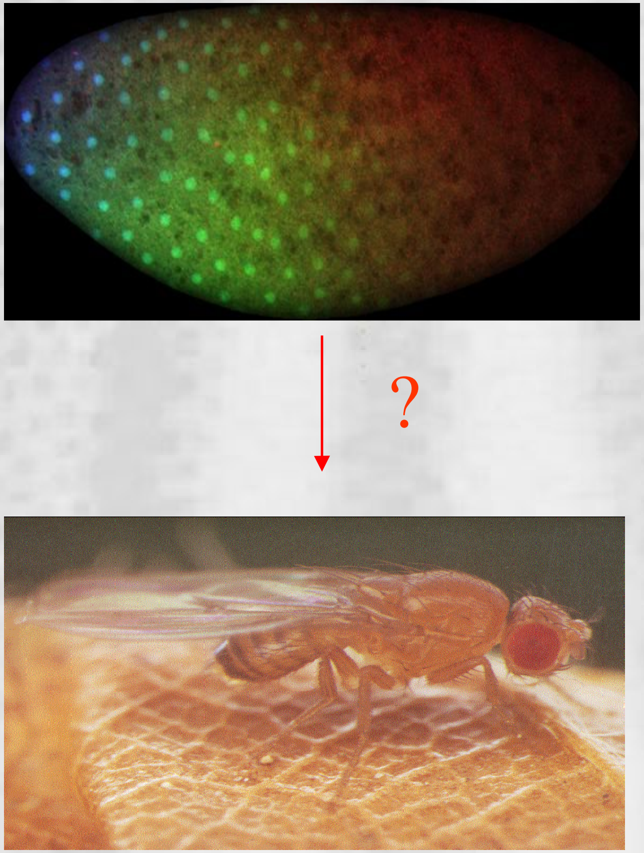


A LIMS to support the analysis of the segment determination system in *Drosophila* early embryo

M.Samsonova, E.Poustelnikova,
A.Pisarev, E. Myasnikova ,
K.Kozlov, J. Reinitz

DeCoBi SPbPU

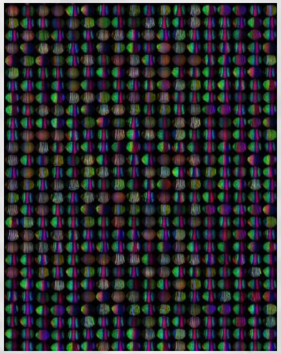
Fundamental Problem of Developmental Biology: Determination of Cell Fate

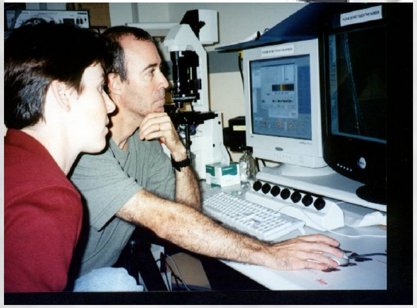


How do different cells of an organism arise, if all of them contain the same genetic material?

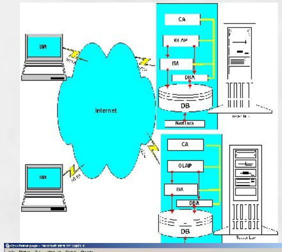
We use a fruit fly as model organism.

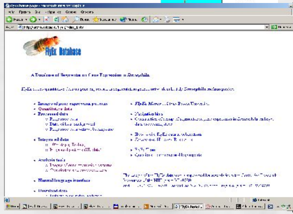
Distributed lab: how do cells define their fate





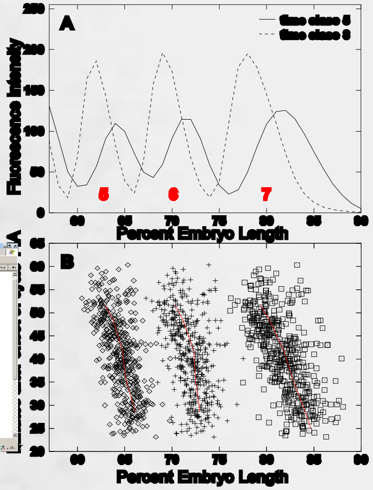
SUNY & UCSD, USA



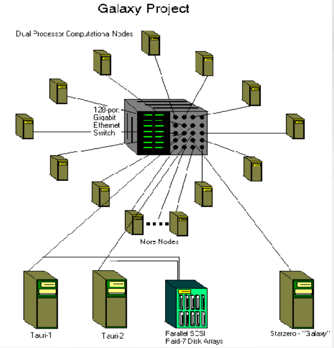


FlyEx Database

- Images of gene expression patterns
- Quantitative data
- Processed data
- Integrated data
- Analysis tools
- Natural language interface
- Download data



SPbPU & FTI, Russsia



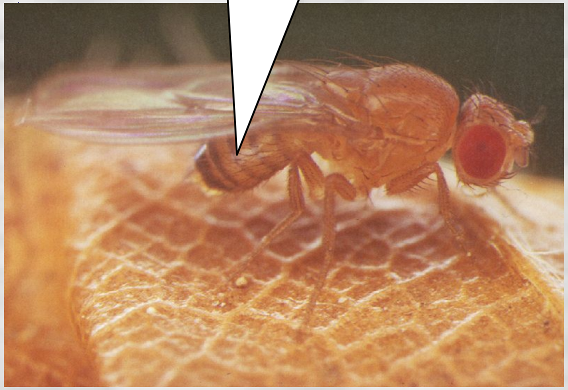
LANL, USA & UvA, Holland

Russian Project Team

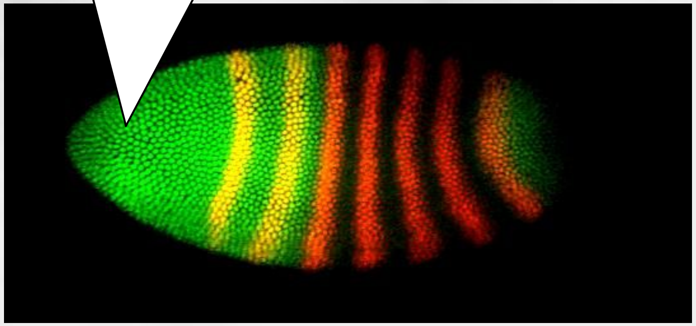
Outline

1. Subject domain and information flow.
2. Requirements to data processing and analysis.
3. System architecture.
4. Implementation:
    - User Interface.
    - Real-life scenarios.

Morphogenetic Field Controlling Segmentation in *Drosophila*



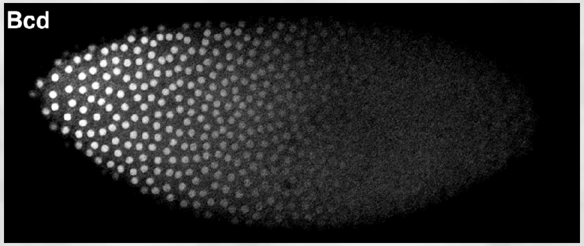
Segments



Morphogenetic field: the entire embryo at the blastoderm stage

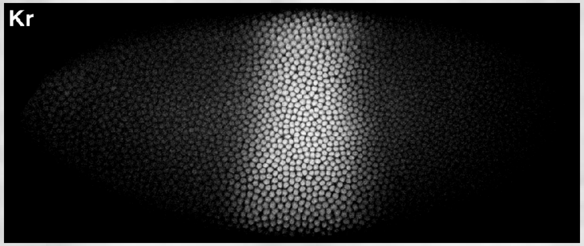
Four Classes of Segmentation Genes which Act in the Blastoderm

Maternal Coordinate Genes



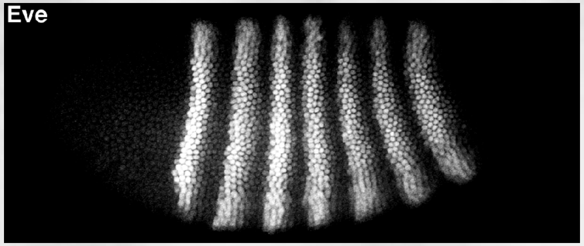
bicoid (bcd)
caudal (cad)
hunchback (hb)

Gap Genes



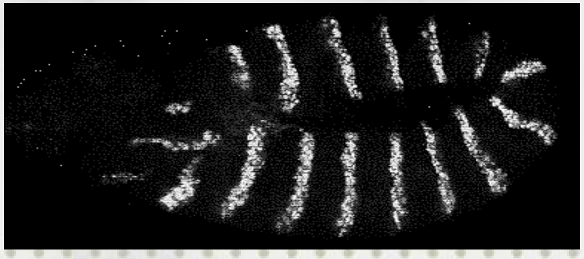
hunchback (hb)
Krüppel (Kr)
knirps (kni)
giant (gt)
tailless (tll)

Pair-Rule Genes

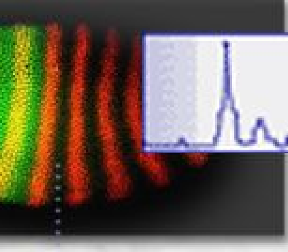


even-skipped (eve)
odd-skipped (odd)
hairy (h)
runt (run)
fushi-tarazu (ftz)
paired (prd)

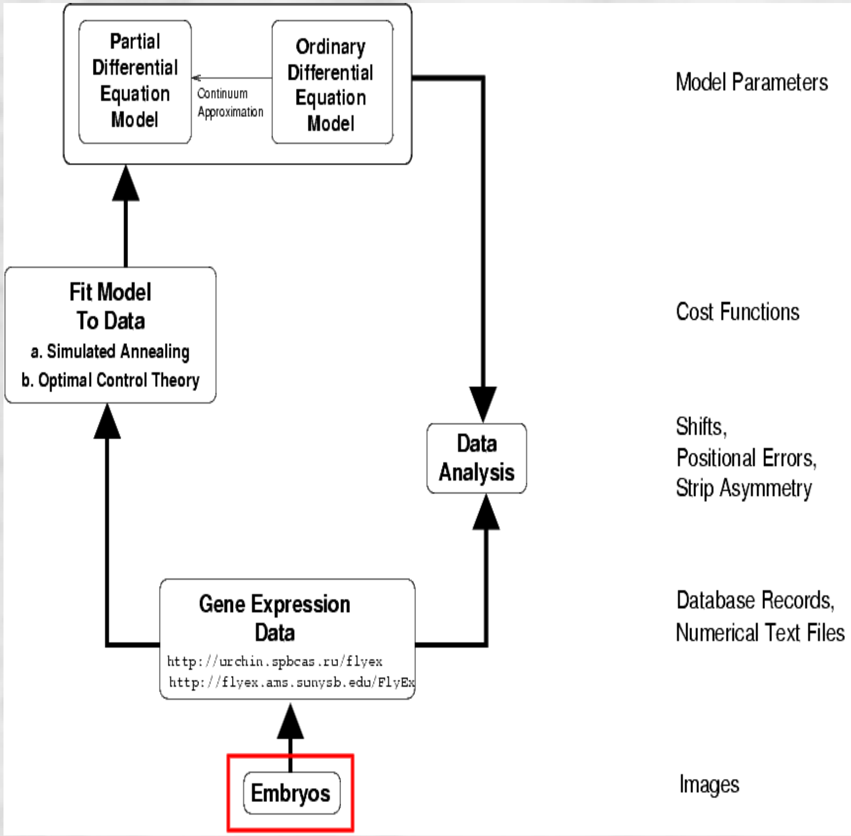
Segment Polarity Genes



engrailed (en)
wingless (wg)
sloppy-paired (slp)



Workflow: from embryo images to data analysis



Partial Differential Equation Model

Continuum Approximation

Ordinary Differential Equation Model

Fit Model To Data

a. Simulated Annealing

b. Optimal Control Theory

Data Analysis

Gene Expression Data

http://urchin.spbcas.ru/flyex

http://flyex.ams.sunysb.edu/FlyEx

Embryos

Model Parameters

Cost Functions

Shifts, Positional Errors, Strip Asymmetry

Database Records, Numerical Text Files

Images

Nature, v 430, 15 July 2004: 368-371

letters to nature

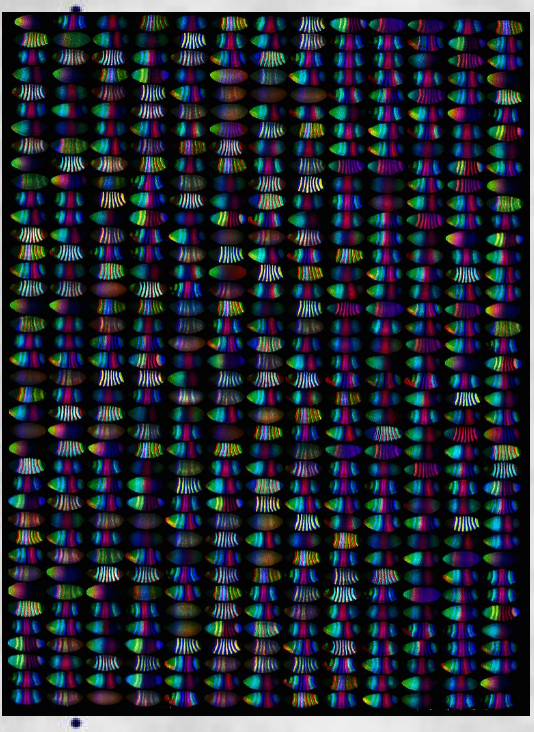
Dynamic control of positional information in the early *Drosophila* embryo

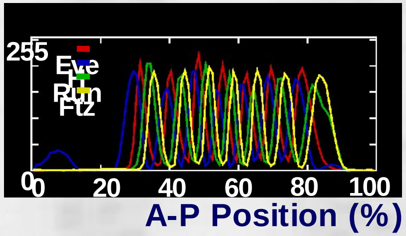
Johannes Jaeger[1], Svetlana Surkova[2], Maxim Blagov[2], Hilde Janssens[1], David Kosman[3], Konstantin N. Kozlov[2], Manu[1], Ekaterina Myasnikova[2], Carlos E. Vanario-Alonso[1,4], Maria Samsonova[2], David H. Sharp[5] & John Reinitz[1]

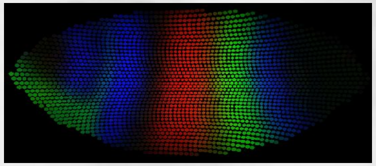
[1] Department of Applied Mathematics and Statistics, and Center for Developmental Genetics, Stony Brook University, Stony Brook, New York 11794-3600, USA
[2] Department of Computational Biology, Center for Advanced Studies, St Petersburg State Polytechnic University, St Petersburg, 195251 Russia
[3] Department of Biology, University of California, San Diego, California 92093, USA
[4] Universidade Federal do Rio de Janeiro, Instituto de Biofísica Carlos Chagas Filho, Rio de Janeiro, Rio de Janeiro 21949-900, Brazil
[5] Applied Physics Division, and Theoretical Division, Los Alamos National Laboratory, Los Alamos, New Mexico 87545, USA

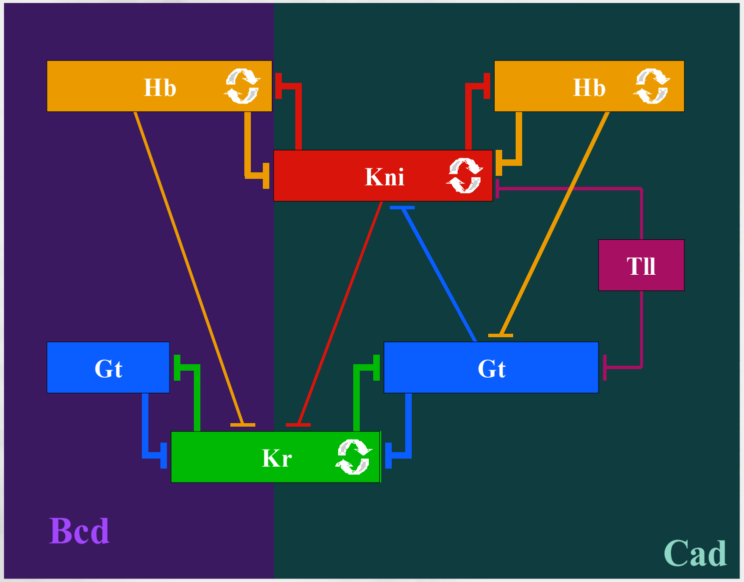
Morphogen gradients contribute to pattern formation by determining positional information in morphogenetic fields[1,2]. Interpretation of positional information is thought to rely on direct, concentration-threshold-dependent mechanisms for establishing multiple differential domains of target gene expression[1,3,4]. In *Drosophila*, maternal gradients establish the initial position of boundaries for zygotic gap gene expression, which in turn convey positional information to pair-rule and segment-polarity genes, the latter forming a segmental prepattern by the onset of gastrulation[5–7]. Here we report, on the basis of quantitative gene expression data, substantial anterior shifts in the position of gap domains after their initial establishment. Using a data-driven mathematical modelling approach[8–11], we show that these shifts are based on a regulatory mechanism that relies on asymmetric gap–gap cross-repression and does not require the diffusion of gap proteins. Our analysis implies that the threshold-dependent interpretation of maternal morphogen concentration is not sufficient to determine shifting gap domain boundary positions, and suggests that establishing and interpreting positional information are not independent processes in the *Drosophila* blastoderm.

Problem statement: From Many Confocal Scans of Embryos Infer the Mechanism of Gene Interactions



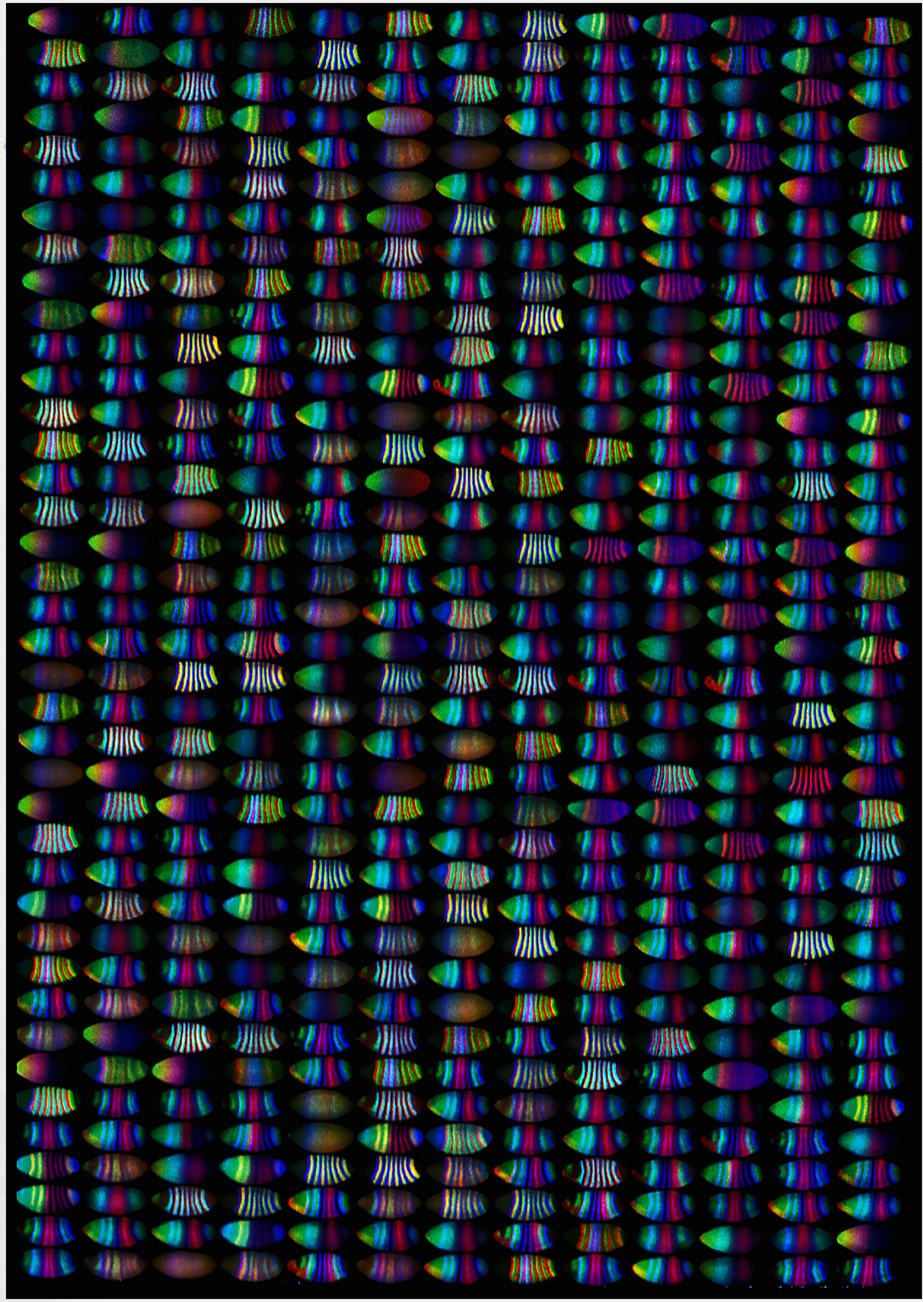


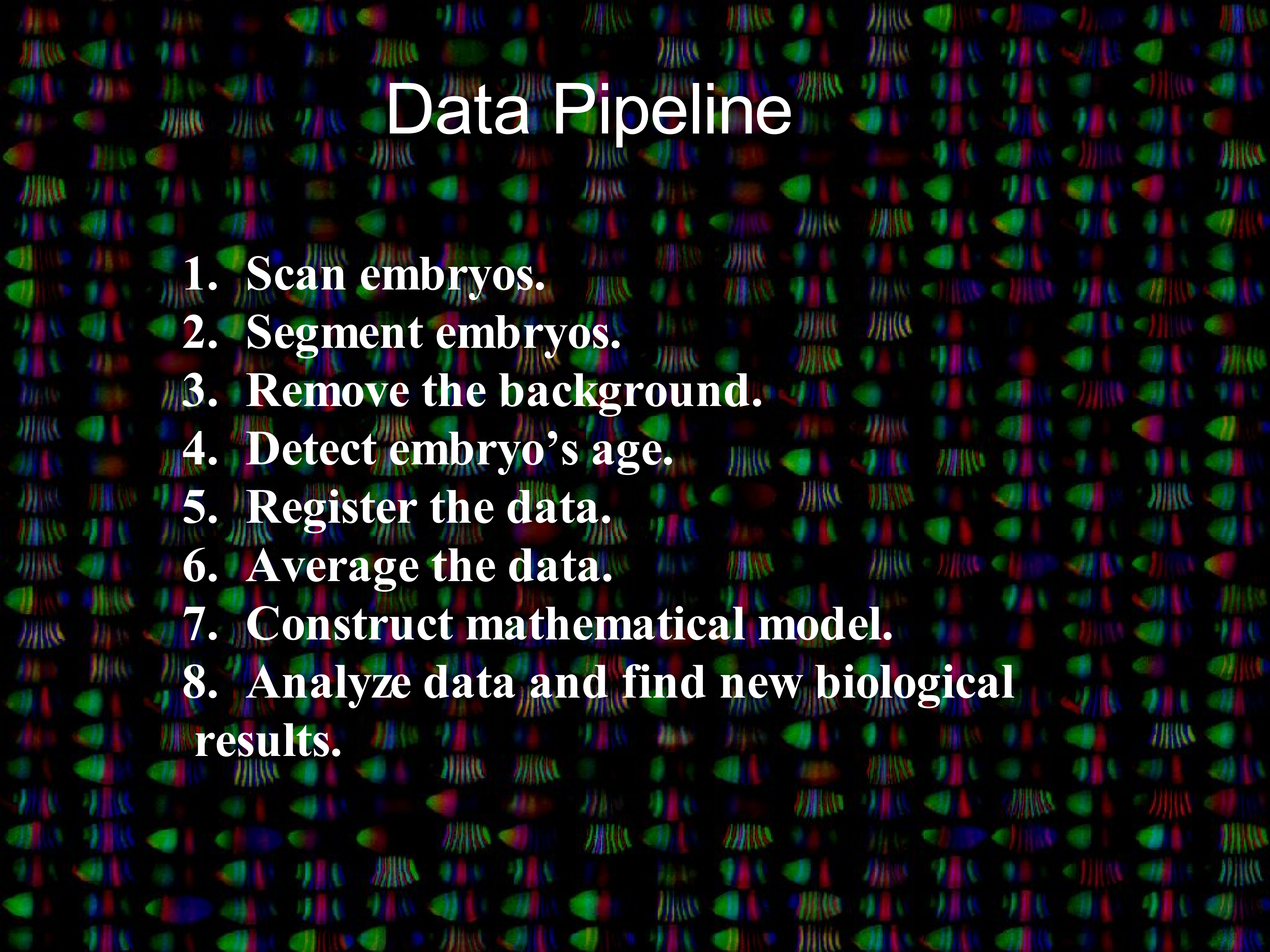




We Start with Many Confocal Scans of Embryos….

(of which these are the first 509)





Data Pipeline

1. **Scan embryos.**
2. **Segment embryos.**
3. **Remove the background.**
4. **Detect embryo’s age.**
5. **Register the data.**
6. **Average the data.**
7. **Construct mathematical model.**
8. **Analyze data and find new biological results.**

PIPE, a Prototype of Laboratory Information Management System: Requirements

✓ provision of fast and simultaneous access of multiple users to shared data and methods;

✓ extendability;

✓ support of actuality and integrity of data;

✓ flexibility in specification and modification of analysis methods;

✓ support of distributed processing and analysis of data;

✓ portability;

✓support of autonomous task performance and notification about results;

PIPE, a Prototype of Laboratory Information Management System: Requirements (Cntd.)

✓ use of heterogeneous software/hardware platforms;

✓ provision of access through FireWall and и Proxy servers;

✓ no need in programming skills or familiarization how to install special software libraries and program tools;

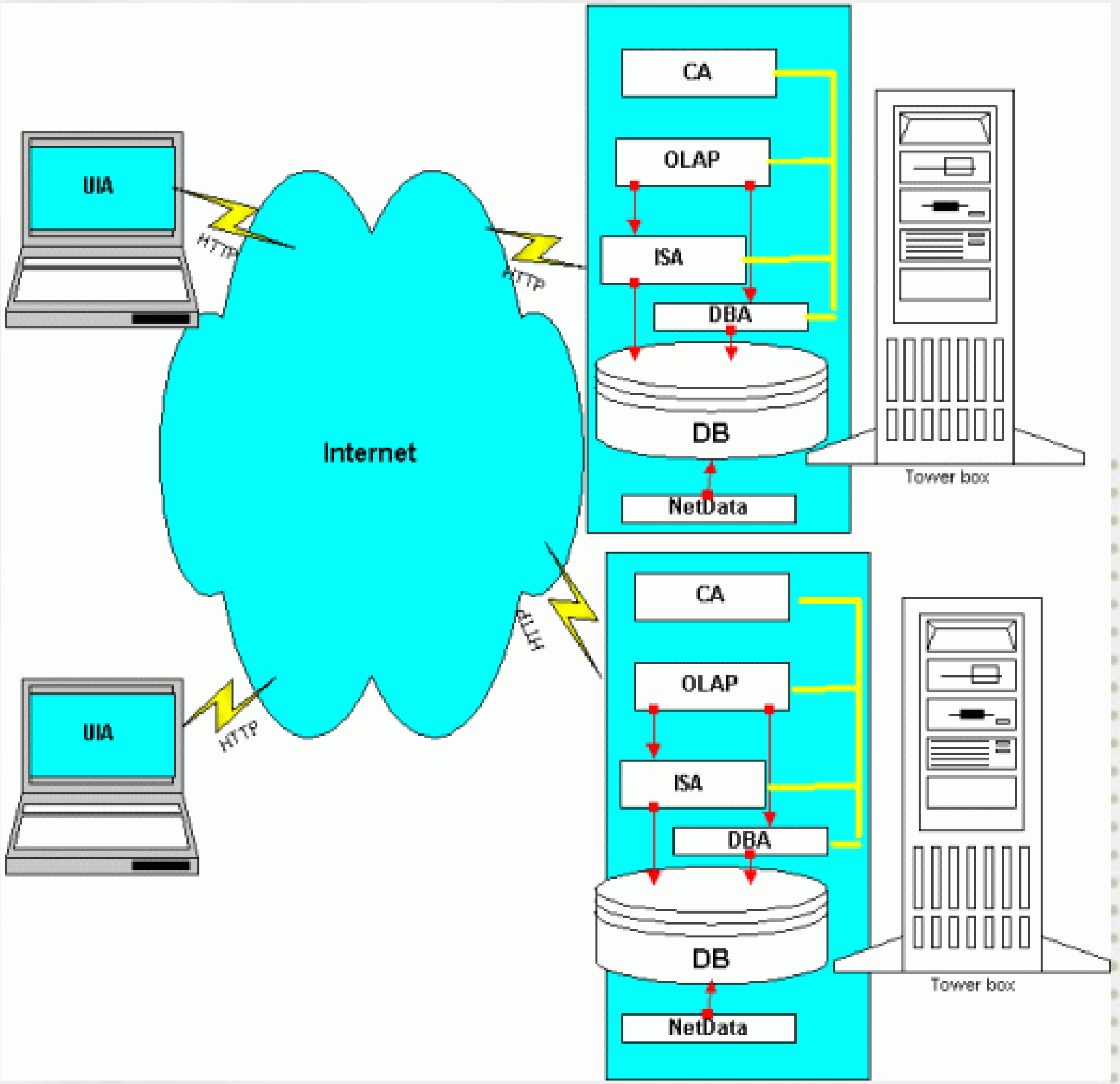
✓ availability of Web-based user interface and visualization tools;

✓ provision of continuous work, sufficient response time and readiness characteristics;

✓ failure-resistance,

✓ preferably based on open source software.

PIPE Architecture



PIPE is developed with standard Java tools

✓ J2SDK Java 2

✓ JavaServer Web Development Kit (JSWDK)

✓ Java Web Services Developer Pack (JWSDP)

✓Apache XML-RPC

Image server agent (ISA)

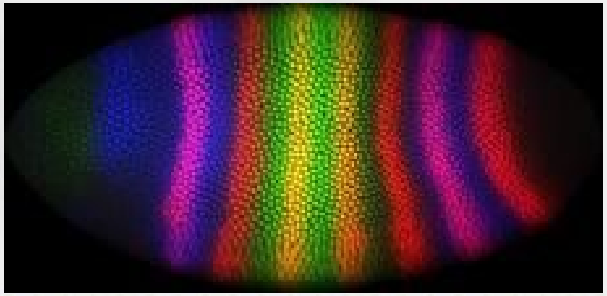
http://urchin.spbcas.ru/downloads/IS/IS.htm

- ✓ interacts with a database via JDBC protocol by use of the XML templates.
- ✓ basic operations on images are implemented using JMagic and ImageMagic packages;
- ✓ subject domain oriented methods for image processing (registration and background removal) are implemented;
- ✓ retrieves images from the DB;
- ✓ enables visualization of processed images as JEPGs.

ISA can be flexibly configured to connect to a database

```
- <XML>- <SOURCE> <DO>getImageFromDB</DO>
  <DESCRIPTION>Information from this TEMPLATE file provides connection
and SQL to relational Data Base</DESCRIPTION>
  <JDBC_DRIVER_CLASS>com.mysql.jdbc.Driver</JDBC_DRIVER_CLASS>
  <DB_URL>JDBC:mysql://hedgehog.spbcas.ru/test</DB_URL>
<DB_USER_NAME>*</DB_USER_NAME>
<DB_PASSWORD>*</DB_PASSWORD>
<COMMENT>/*image- … */ /*[table] - …/ /*[idName] - …. */ /*[id] -….
*/</COMMENT>
<SQL>SELECT image FROM [table] WHERE [idName] = [id]</SQL>
</SOURCE>  </XML>
```

http://urchin.spbcas.ru:7771/image?
table1=images&id_name1=id&image_id1=1&table2
=images& id_name2=id& image_id2=2&
table3=images& id_name3=id& image_id3=3&
compose=combine& zoomw=0.20&zoomh=0.20



OLAP agent

- ✓ cooperates with DBA and ISA to execute complex scenarios by implementing logical rules;
- ✓ communicates with local database via JDBC and remote database via DBA;
- ✓ interacts with registered workflow modules providing for their initialization, function calls and result output.

CA agent

- supports agent registration;
- notifies agents about the current system status in respond to their request;
- monitors the functionality of the system;
  notifies the registered agents about agent failure and other changes in the configuration of the system;
- notifies the system administrator about these changes by e-mail.

Monitoring of the PIPE functionality

```
<xml><email>pisarev@spbcas.ru</
email><agentname>Venus</agentn
ame><theme>Flyex.ams functional
monitoring</theme><diagnosis>
		<rules>
$flyex.ams=DBA.Counter|DBA|
DBA.db2|ams.is|ams.Netdata|
ams.www;
...
$DB2=DBA.db2|ams.is|
ams.Netdata;
$JDBC=DBA.db2|ams.is;
		</rules>
		<text>
$flyex.ams=Flyex.ams.sunysb.edu;
$DBA=Data Base Agent;
$IS=Image Server;
$DB2=IMB DB2;
$JDBC=JDBC DB2;
		</text>
</diagnosis>
```

```
<test>
		<n>1</n>
		<name>dba.Counter</name>
		<url>
http://flyex.ams.sunysb.edu:7005/counter?fl
yex.jpg
		</url>
		<copy>
MasterCopy/urchin/Counter/counter.jpg
		</copy>
…
		<explanation>
http://flyex.ams.sunysb.edu:7005/counter?...
is used for check of availability of port=7005
DataBaseAgent service
		</explanation>
</test>
<test>
```

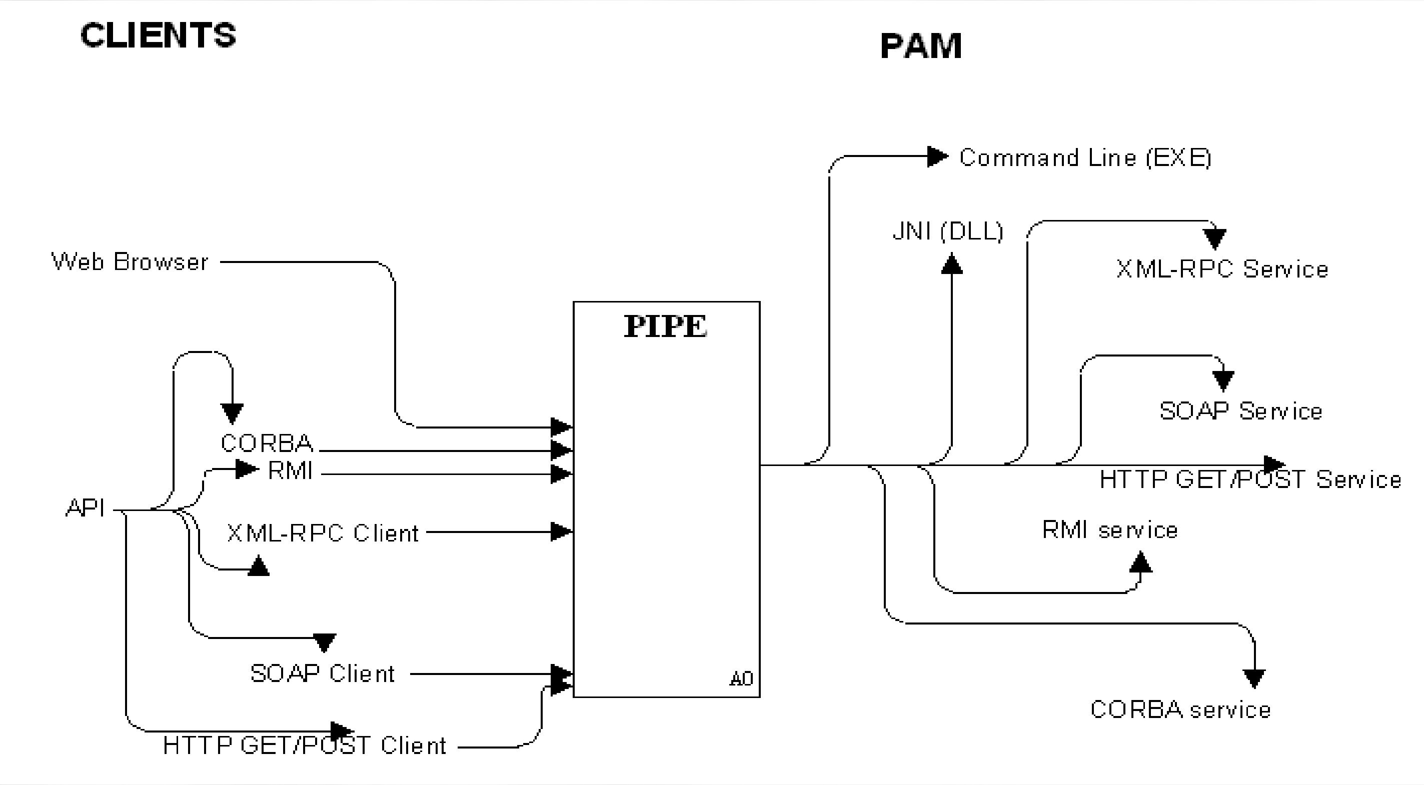
Program modules

In general each workflow consists of many steps, which we call as program modules (PM).

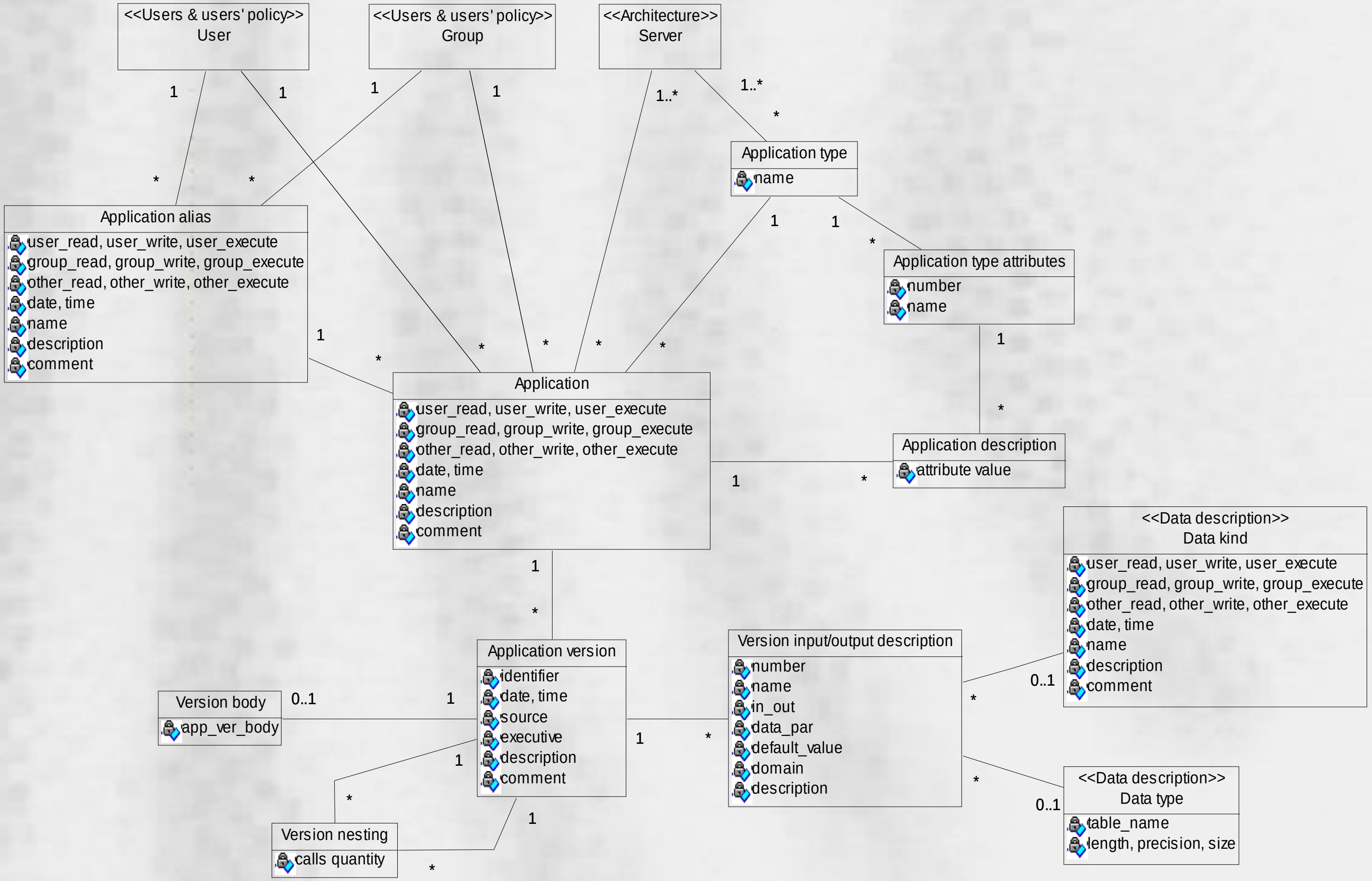
PM communicate with each other via the OLAP server agent.

We implement program modules in C++ or Java.

PIPE interfaces between potential clients, program modules and services



Metadata



<<Users & users' policy>>
User

<<Users & users' policy>>
Group

<<Architecture>>
Server

Application type
- name

Application alias
- user_read, user_write, user_execute
- group_read, group_write, group_execute
- other_read, other_write, other_execute
- date, time
- name
- description
- comment

Application type attributes
- number
- name

Application
- user_read, user_write, user_execute
- group_read, group_write, group_execute
- other_read, other_write, other_execute
- date, time
- name
- description
- comment

Application description
- attribute value

<<Data description>>
Data kind
- user_read, user_write, user_execute
- group_read, group_write, group_execute
- other_read, other_write, other_execute
- date, time
- name
- description
- comment

Application version
- identifier
- date, time
- source
- executive
- description
- comment

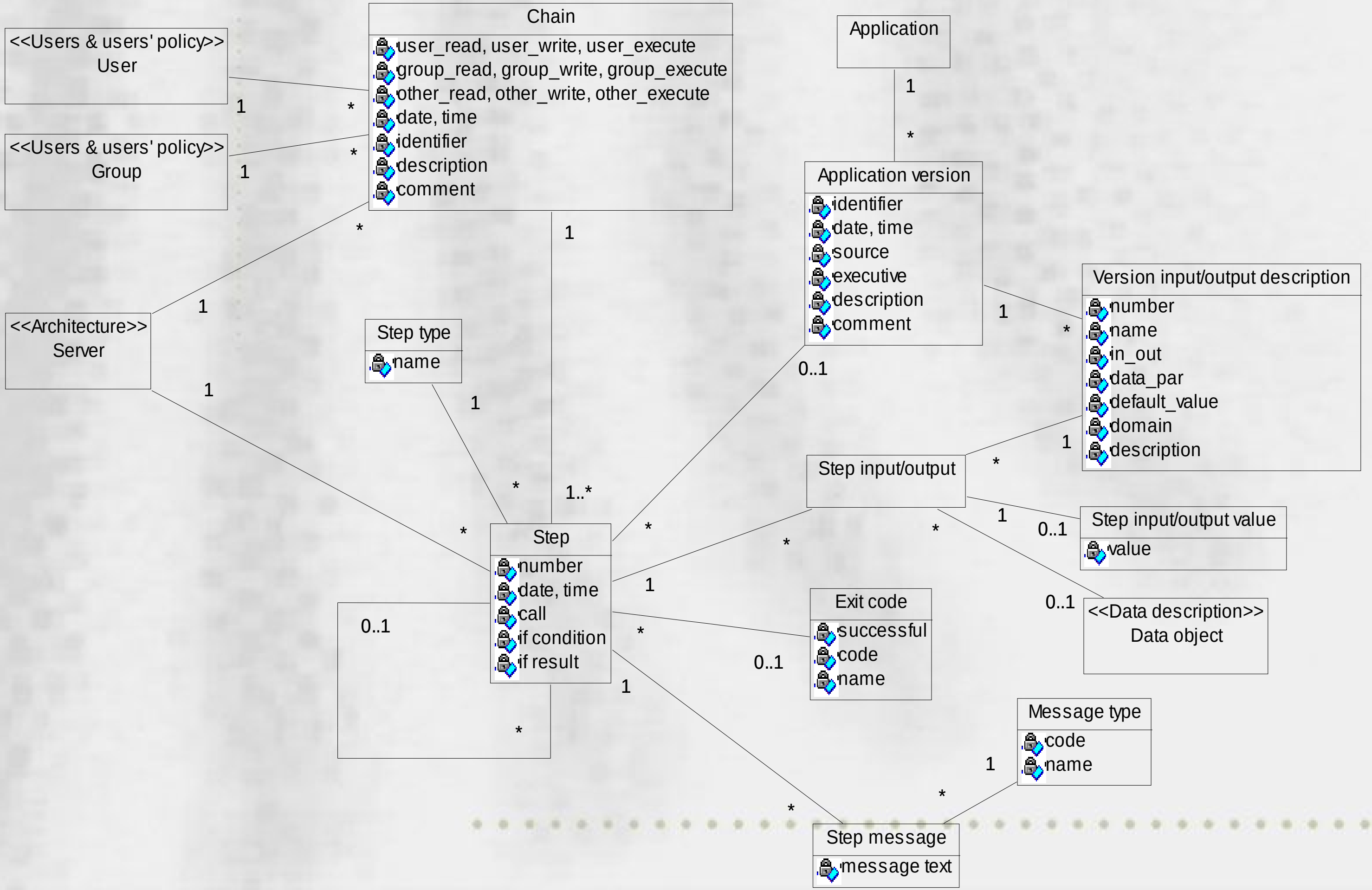
Version input/output description
- number
- name
- in_out
- data_par
- default_value
- domain
- description

Version body
- app_ver_body

Version nesting
- calls quantity

<<Data description>>
Data type
- table_name
- length, precision, size

Metadata



Chain
- user_read, user_write, user_execute
- group_read, group_write, group_execute
- other_read, other_write, other_execute
- date, time
- identifier
- description
- comment

<<Users & users' policy>>
User

<<Users & users' policy>>
Group

<<Architecture>>
Server

Step type
- name

Step
- number
- date, time
- call
- if condition
- if result

Application

Application version
- identifier
- date, time
- source
- executive
- description
- comment

Version input/output description
- number
- name
- in_out
- data_par
- default_value
- domain
- description

Step input/output

Step input/output value
- value

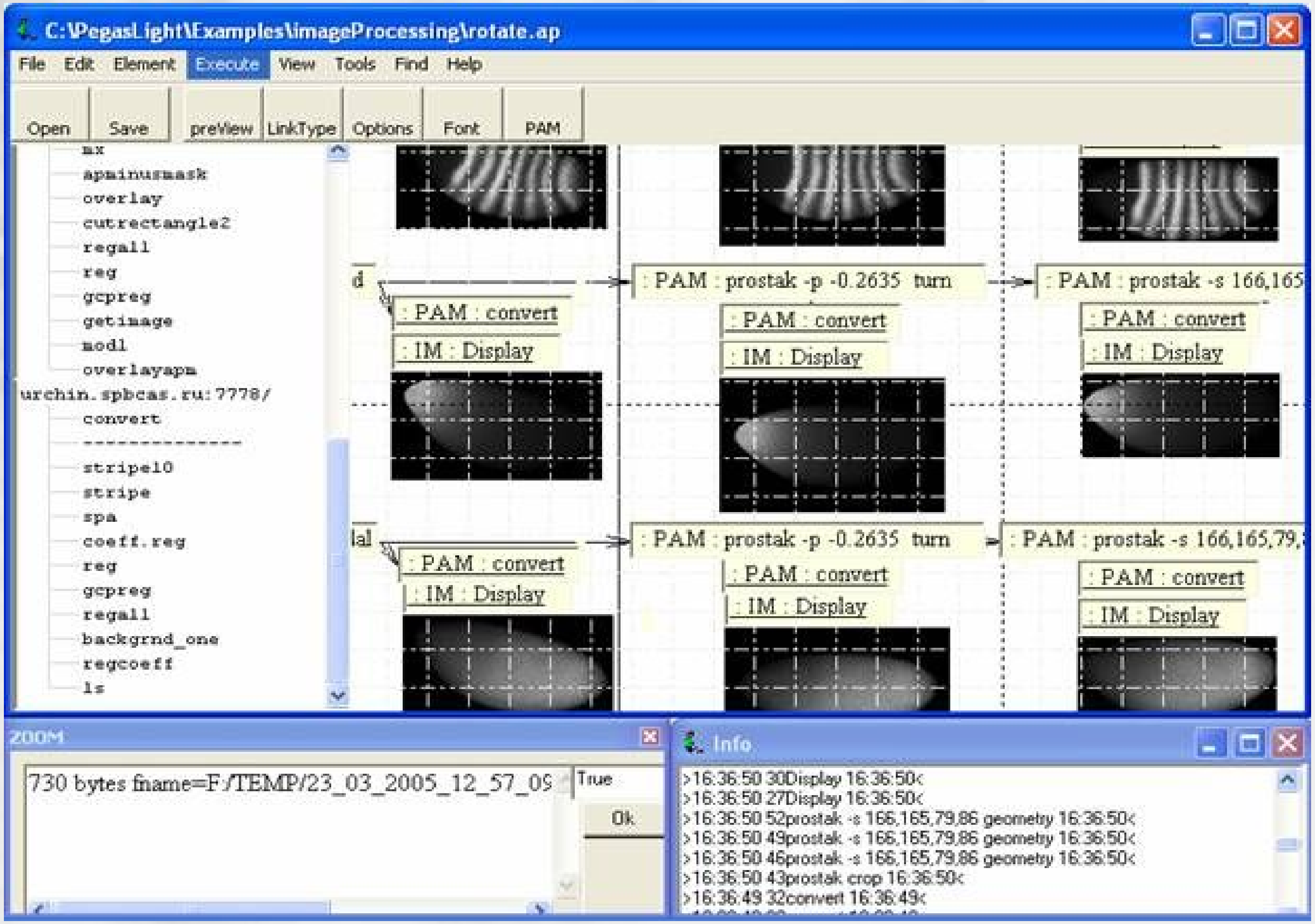
<<Data description>>
Data object

Exit code
- successful
- code
- name

Message type
- code
- name

Step message
- message text

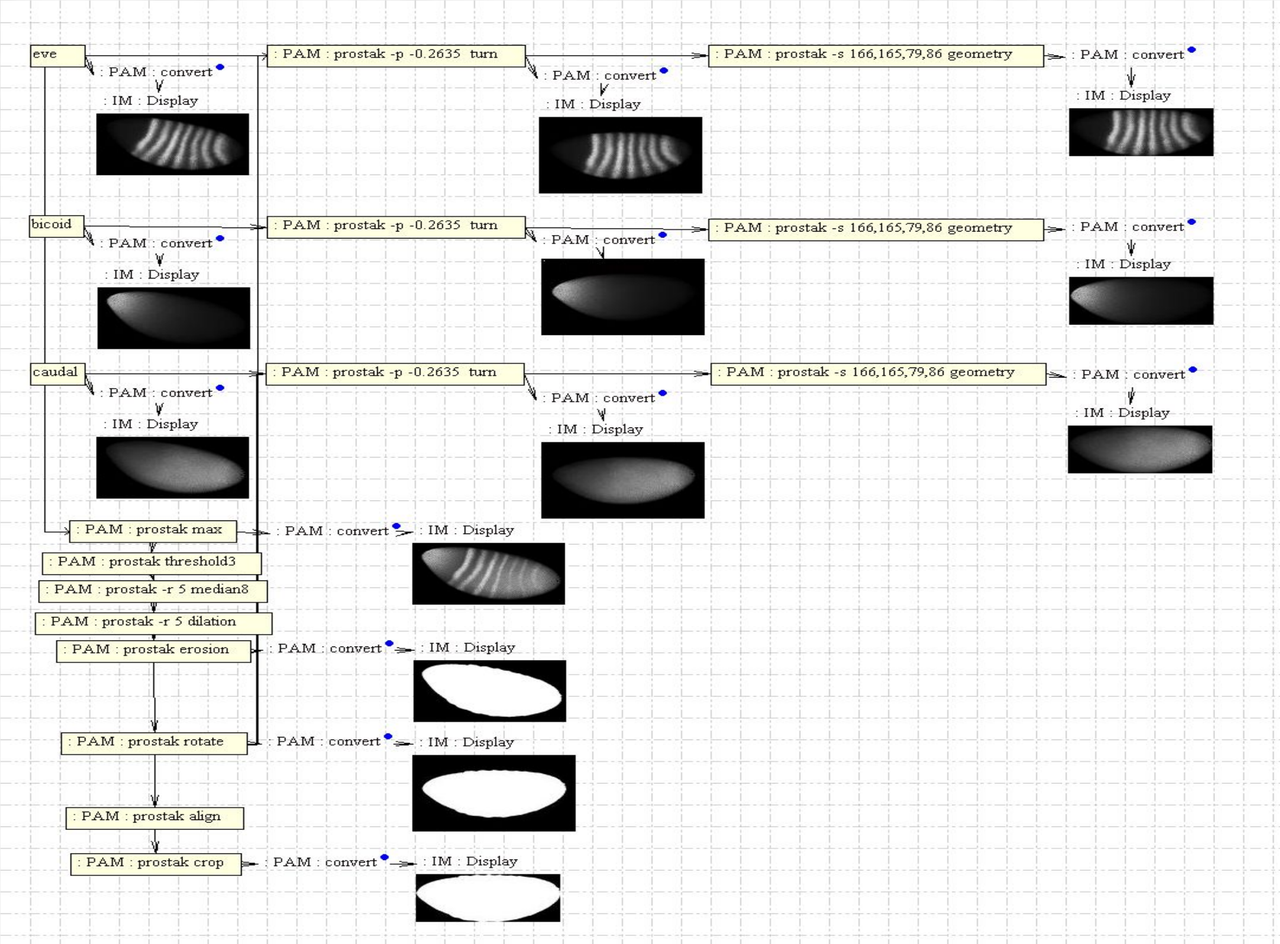
PIPE in Use: Interface



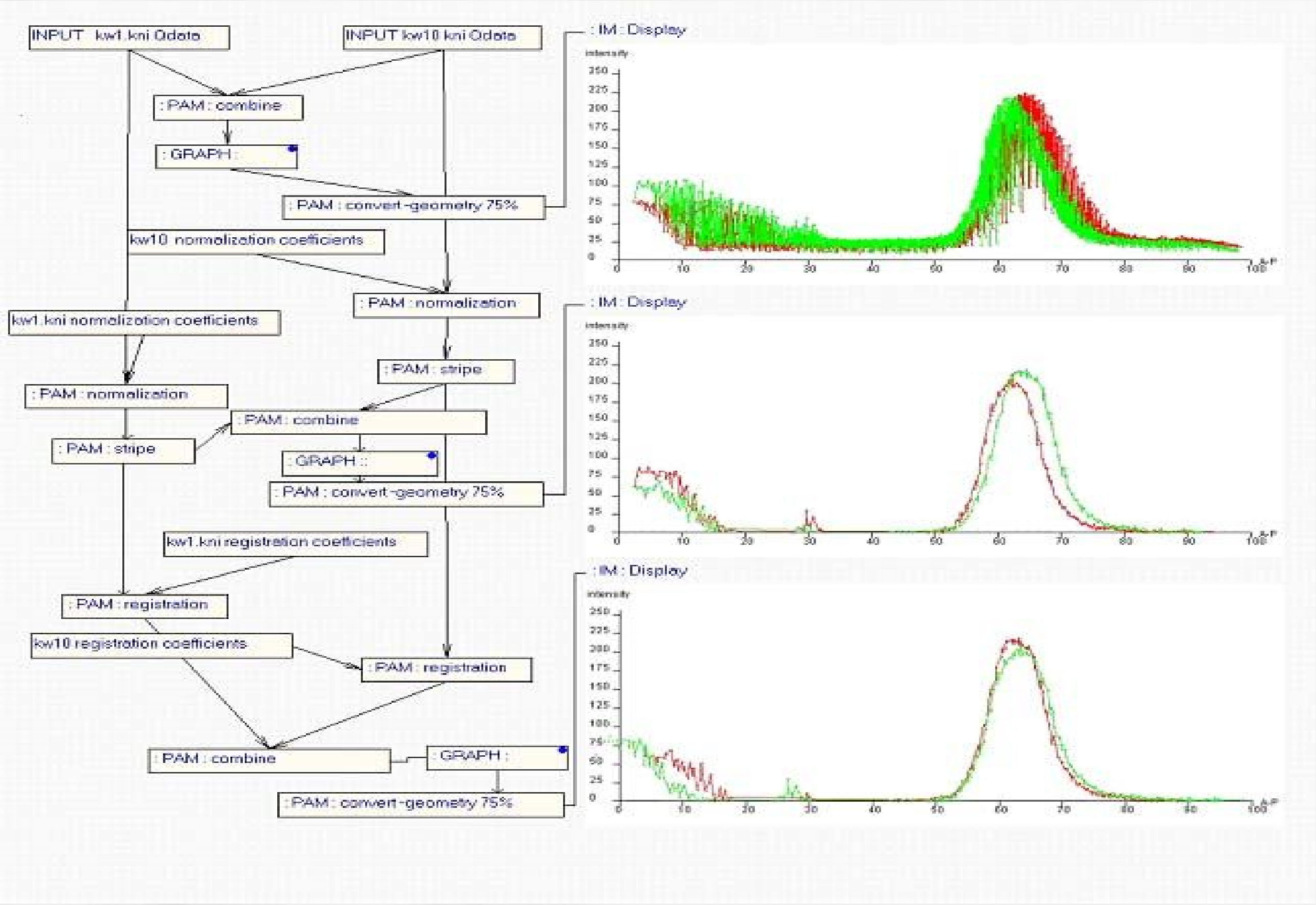
730 bytes fname=F:/TEMP/23_03_2005_12_57_09

```
>16:36:50 30Display 16:36:50<
>16:36:50 27Display 16:36:50<
>16:36:50 52prostak -s 166,165,79,86 geometry 16:36:50<
>16:36:50 49prostak -s 166,165,79,86 geometry 16:36:50<
>16:36:50 46prostak -s 166,165,79,86 geometry 16:36:50<
>16:36:50 43prostak crop 16:36:50<
>16:36:49 32convert 16:36:49<
```

PIPE in Use: Rotating and Flipping of an Embryo Image



PIPE in Use: Processing of Quantitative Gene Expression Data



Advantages of the PIPE architecture

- ✓ adaptability;
- ✓ possibility to extend to interact with XML/RPC, SOA, OGSA, RMI, CORBA applications;
- ✓ effective processing of images in standard formats;
- ✓ fault tolerance;
- ✓ easy integration of already developed programs.

Conclusions:

Our approach results in

- ✓ regulation, standardization and automation of data processing and analysis procedures,
- ✓ reduction of data processing errors,
- ✓ acceleration of system reactivity to user requests,
- ✓ effective use of expensive equipment and personnel skills.

This increases efficiency and productivity of work, as well as expand potentialities of all performance sites to analyze and process data.