Introducing python into industrial environment applications

Fabio Pliger

SIA s.r.l.

fabio.pliger@siavr.it

Industrial Supervision and Scada Frameworks

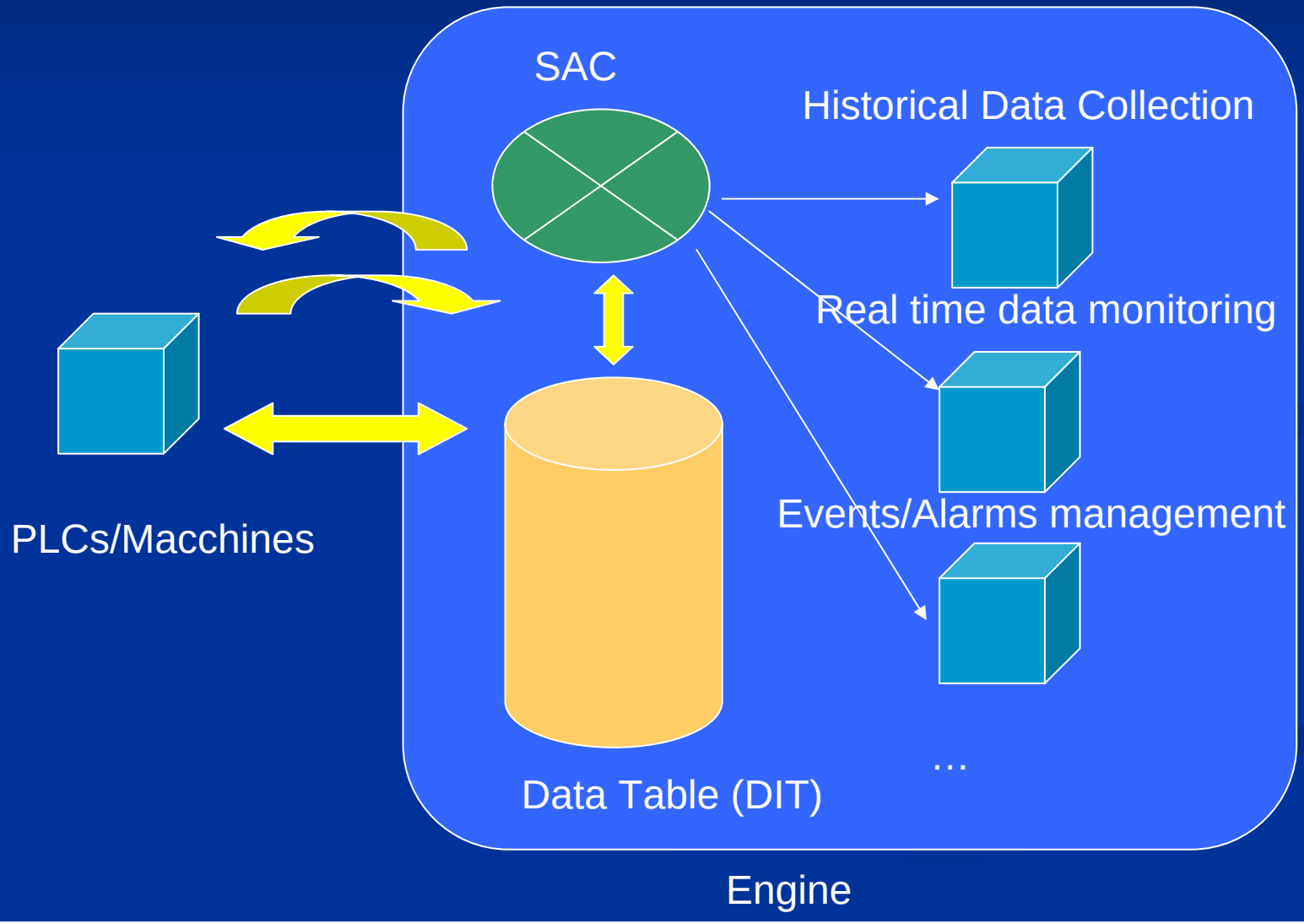
- process visualization, data acquisition and supervisory control
- allow to control and monitor productive units around
- Client/Sever distributed and open architecture
- Real Time Information Portal

Industrial Supervision and Scada Frameworks(2)

- Collect “quality” data (both analogic and digital)
- Alarms/Events management
- consistent performance and known response times
- audit trails in order to maintain a tight control over their operation
- security and accountability

BRIEF ARCHITETURE SCHEME

(really really semplified..for iFix)



We need more…

- Scada frameworks are great but people need more and more information. So, we need new applications that can perform those tasks.
- + features = + $$$$$$
- Our cusotmers loves to have personal solutions! They feel they are important.. ☺

… is python what we need?

- Before 3 years ago applications were always C and VB…
- 3 years ago: can we do better?
- Hey, this python language.. Looks realy cool!

… Why python?

- It's cross platform (???)
- time reduction developing new projects
- Debug is easy
- Dynamic typing
- …but we need 2 different aproaches:
  1. Let's use it now(!) for new applications with a deadline already fixed
  2. Oh.. I think it's better if we stress and test python before we use in our "market" applications

Aproach 1

We decided this aproach when we had particular situations like:

- No high performance needs
- Time is running low… fast development is needed!
- Other languages didn't satisfy us…

Aproach 1 - Results

- We used python to develope some data plotting applications.
- Chaco il great!
- It was “easy” to wrap SCADA dlls and access/manage to its data from python
- It was easy extend, manipulate and add information to scada data.

Aproach 1 – Results(2)

- We had a new powerful application
- customer was so satisfied with that application that suddenly asked us to add new features
- Ok! We are ready to try it for other apps!

Aproach 1 – The Batch Report

- production batch cycle can be really complex
- it is important to track these production processes (talking about pharmaceutical companies it's a "must")
- Scada frameworks are useful but poor for batch tracking…

Aproach 1 – The Batch Report Generator

- we did is a high classification of the structure of all the production system and its actors, variables, subsystems…
- extra data is stored in a database outside the scada database and defines all the connections between machines, tanks, sensors, variables…
- map objects in the production process with their respective variables in the scada database
- program can run outside the scada system, monitor what is happening and collect data
- digitals and alarms are event based and analog variables are sampled at a predefined frequency >> time strategies are required!
- Different machines can mean different data structures

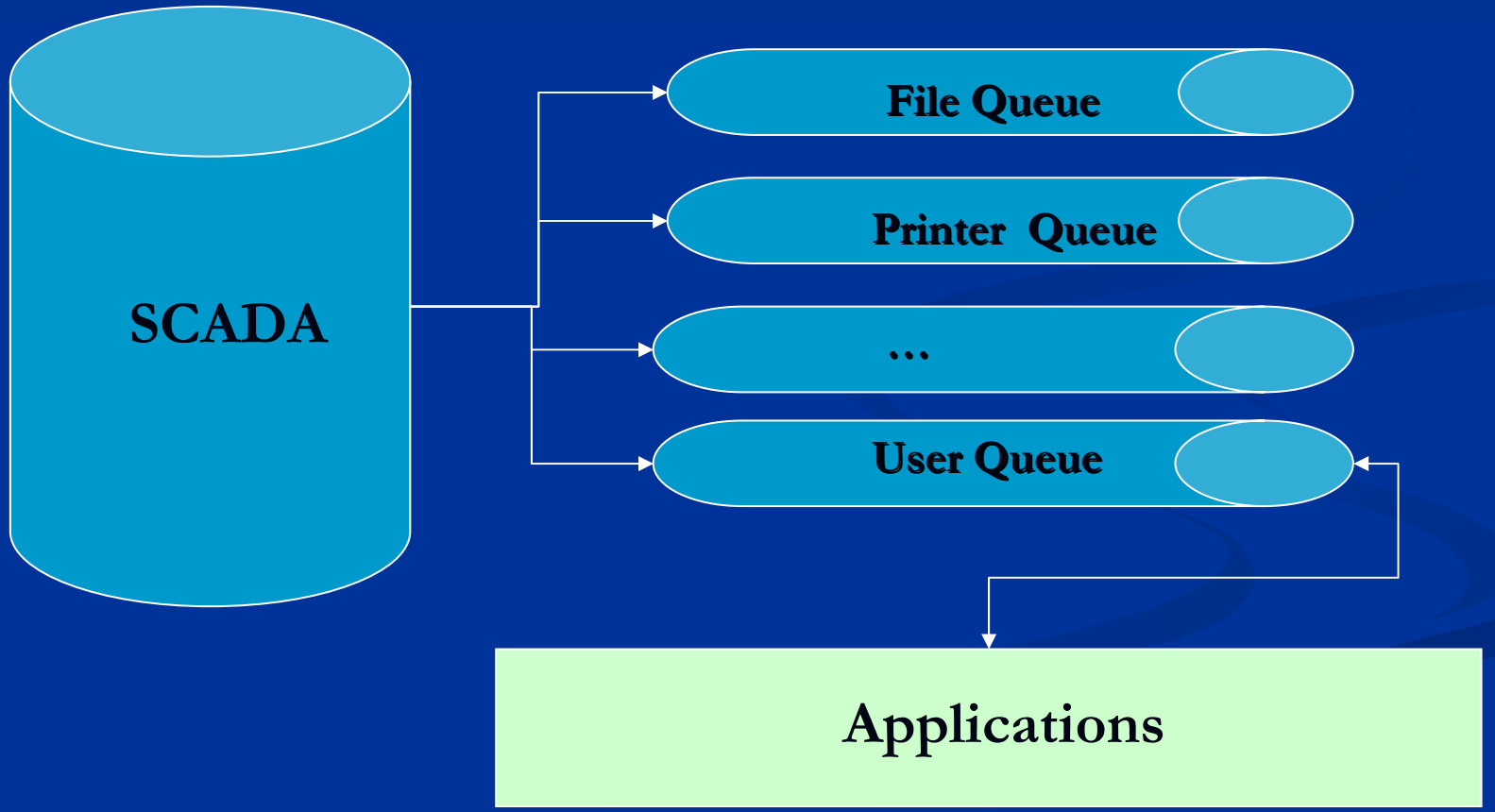
Aproach 1 – The Batch Report Generator - Results

- data is assigned to each machine that generated it
- each phase of each machine is tracked (for scada system events are just digital type variables)
- every alarm called during the batch is tracked and assigned
- the user can see, in the report, if all the process is ok or it had any problem
- few seconds/minutes after the scada system has finished registering the production batch, the program is able to generate reports of the machines batch, connect them and print it all.
- It can show alarms, analog variables linked to any machine, customize them and remake the report with any configuration the user wants.

Aproach 2

- We decided to use this aproach in all those cases we needed a really stable application with high performance needs.
- We decided to create basic modules and then add new features (other modules) when we needed
- The 1° effort was done to wrap iFix (scada) dlls and use them with python (this was useful also for aproach 1!)
- Then we started testing it for an application that comunicates with the scada framework and manage alarms/events generated by it and for an app that downlownd buffered data buffered in plc buffered and syncronises itself with it.

Aproach 2 (2) - Events



Aproach 2 - Results

- Comparing with similar c applications performance python was not that slow… (only in some rare cases but we can do better…)
- We need to use threads…
- Python is much, much, much more flexible!
- Probably we'll dedicate sources to improve what we obtained

Conclusion

- Python worked well for us
- Python reduced about 50% of time required for development
- UI take much of our time
- Still some pitfalls (apps distribution, track modules dependecies, documentation*)

THANKS