DESIGNING INTERACTION
A MIDDLEWARE APPROACH

http://www.irit.fr/~Philippe.Truillet

v.2.5 – september 2021

A DISTRIBUTED ARCHITECTURE?

Systems become more and more complex

- More (new) devices
- A lot of information
- …
    - → need for distributed applications

Distribution principles: interprocess communication

- upon *socket*
- With more complex abstraction levels (RPC: Remote Procedure Call, RMI : Remote Method Invocation, …)

A DISTRIBUTED ARCHITECTURE?

cons …

- need to know where are the objets (which address?)
- important learning cost
- specific frameworks (ex: RMI, OSGi, …)
- execution and architecture models are inconsistent

A DISTRIBUTED ARCHITECTURE FOR HCI?

actually, few middlewares are « interaction-oriented »

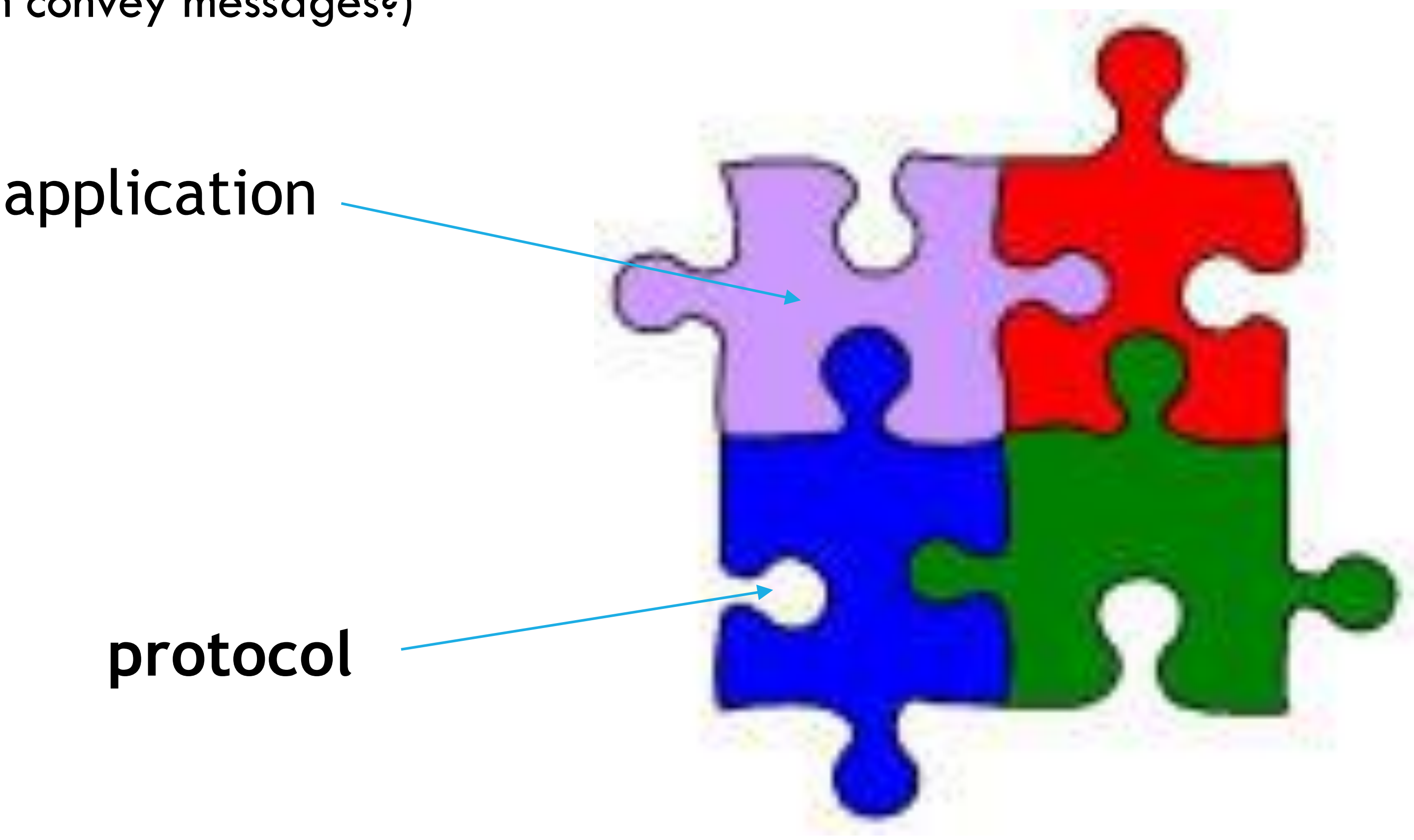
what are the needs?

- Separation interface/functional core
- sending and receiving events, not method invocation!

→ a (possible) solution: an « **event-oriented** » middleware

A DISTRIBUTED ARCHITECTURE FOR HCI?

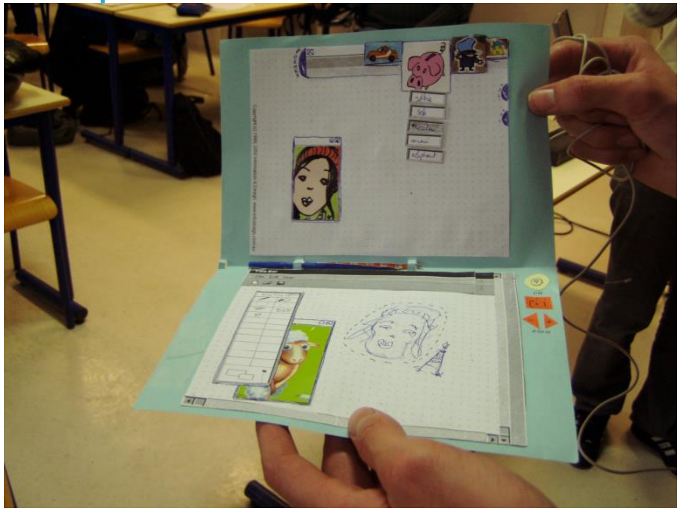
the most important thing is to define the protocol (what kind of information convey messages?)

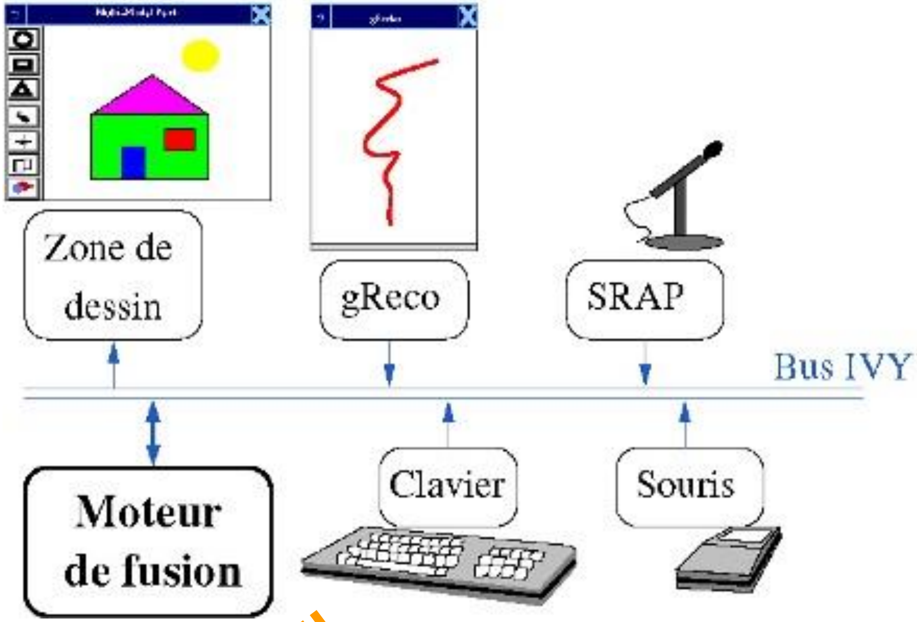


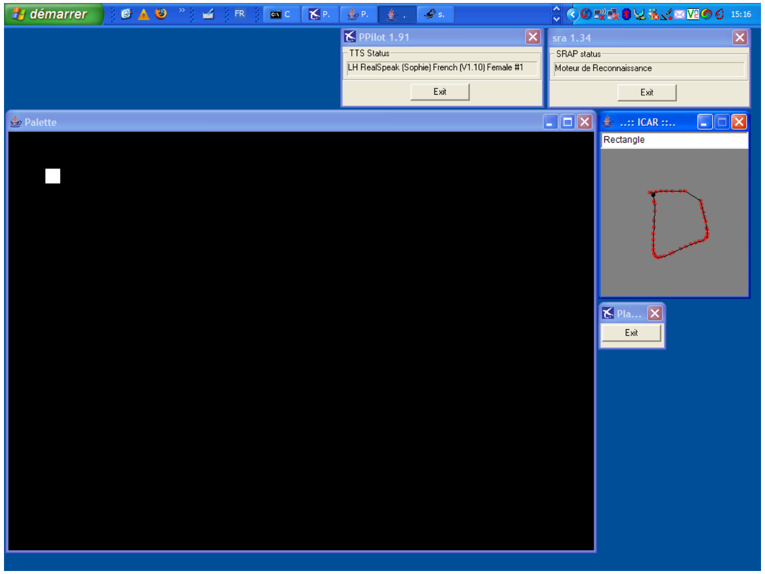
application

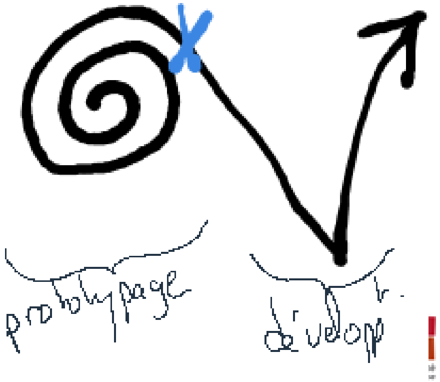
protocol

OBJECTIVES




Zone de dessin
gReco
SRAP
Bus IVY
Moteur de fusion
Clavier
Souris


démarrer
FR
15:16
PPilot 1.91
TTS Status
LH RealSpeak (Sophie) French (V1.10) Female #1
Exit
sra 1.34
SRAP status
Moteur de Reconnaissance
Exit
Palette
..:: ICAR ::..
Rectangle
Pla...
Exit


prototypage
développt.

OBJECTIVES

during the design process

- modularity = re-usability
- You can use many frameworks and languages
- rapid-prototyping from **paper** to **hi-fi**

during the testing phase

- ability to test different modules separately

THE IVY MIDDLEWARE

Ivy was created in 1996 in order to prototype highly interactive softwares for civilian aviation purpose.

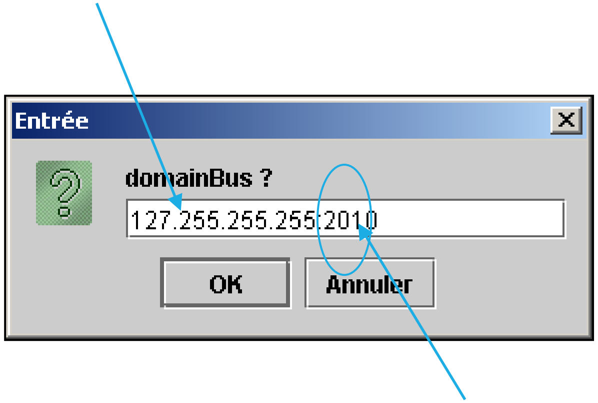
ivy is simple (**http://www.eii.cena.fr/products/ivy)**

- to understand,
- to use
- and it's free! ;-)

ivy is a simple protocol and a set of open-source (LGPL) libraries and programs that allows applications to broadcast information through text messages, with a subscription mechanism based on regular expressions.

THE IVY MIDDLEWARE

IP address
Broadcast address
Multicast address



domainBus ?

127.255.255.255:2010

port

- ivy is not based on a client/server protocol
  - each module proposes services
  - each module reacts to events

- similar to event programming concepts (java, X-window, visual C++, …)

THE IVY MIDDLEWARE

Try regex!

https://regex101.com/

Exchange protocol is *purely textual* (regexp binding)

- examples:
  ICAR command=back

  IMM media=SRAP action=previous

variable/value

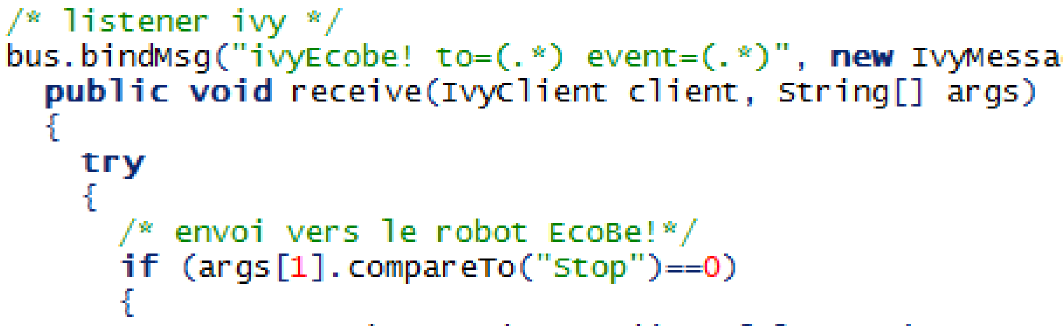
application name

4 main steps:

1. Create a bus object
2. « start the bus »
3. Define messages to send and regexps (in order to receive messages)
4. Stop the bus before leaving

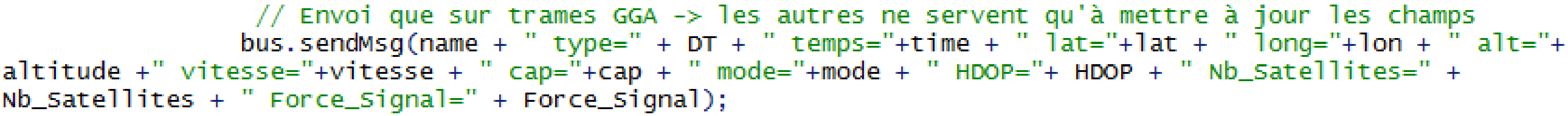
THE IVY MIDDLEWARE

2 « basics » mecanisms : receiving (**bindMsg**) and sending (**sendMsg**) messages



```
/* listener ivy */
bus.bindMsg("ivyEcobe! to=(.*) event=(.*)", new IvyMessageListener() {
  public void receive(IvyClient client, String[] args)
  {
    try
    {
      /* envoi vers le robot EcoBe!*/
      if (args[1].compareTo("Stop")==0)
      {
```

Callback function



```
                    // Envoi que sur trames GGA -> les autres ne servent qu'à mettre à jour les champs
                   bus.sendMsg(name + " type=" + DT + " temps="+time + " lat="+lat + " long="+lon + " alt="+
altitude +" vitesse="+vitesse + " cap="+cap + " mode="+mode + " HDOP="+ HDOP + " Nb_Satellites=" +
Nb_Satellites + " Force_Signal=" + Force_Signal);
```

THE IVY MIDDLEWARE

ivy library is implemented

- in C, C++, C#, java, processing.org, perl, perl/Tk, Tcl, Tcl/Tk, CAML, Ada95, Python, VBA, Flash, COM objects, …
- under MacOS, Win32, WinCE, Windows Mobile, Android, Un*x, linux, …

consequence: design process become easy.

Designers can use the best language for each problem to address

CONCLUSION

event-oriented middleware allows:

- to focus on design problems and not only on implementation problems
- to rapidly prototype to « *have a look* » and test a (not final) system

KNOWN USERS

Labs

IRIT

INT

dgac

DSNA

ENST Bretagne

CLIPS

Communication Langagière et Interaction Personne-Système

Fédération IMAG

BP 53 - 38041 Grenoble Cedex 9 - France

TRIGONE

ENAC-LII

ENAC

Compagnies

eDF

CURTEK SYSTEMS

intuilab

innovative interfaces for people

THALES

sopra steria

iNGENUITY i/o

EUROCONTROL

UNIVERSITÉ TOULOUSE III PAUL SABATIER

Université de Toulouse

LINKS

ivy official websites

- http://www.eei.cena.fr/products/ivy/download (sometimes broken)
- https://gitpub.recherche.enac.fr/ivy (*Git*)

Libs

- **Python**: https://gitlab.com/ivybus/ivy-python et https://pypi.python.org/pypi/ivy-python
- **C**: https://github.com/esden/ivy-c/
- **Java**: ~~http://lii-enac.fr/~jestin/homepage/software.html~~
- **https://github.com/truillet/ivy**