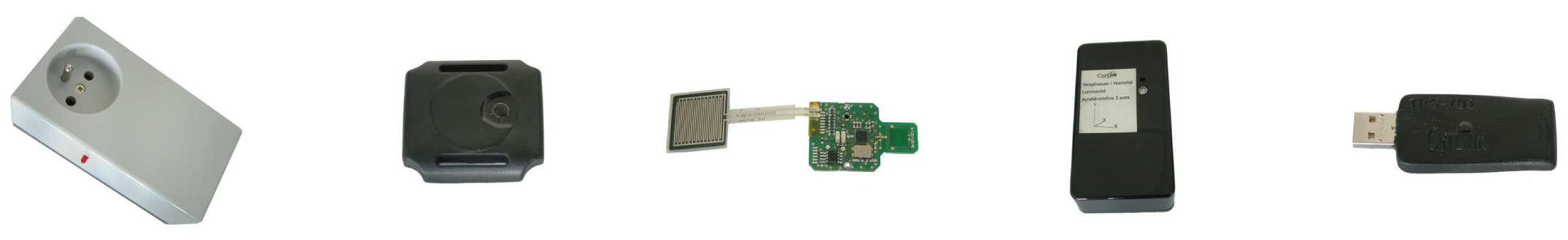
COMPUTER UBIQUARIUM*

A FRAMEWORK FOR THE STUDY OF MOBILE COMPUTERS APPLIANCES IN SIMULATED ENVIRONMENTS

AN ENVIRONMENT CONTAINING SERVICES

• Web Services for Physical Devices

Real Environment: Sensors

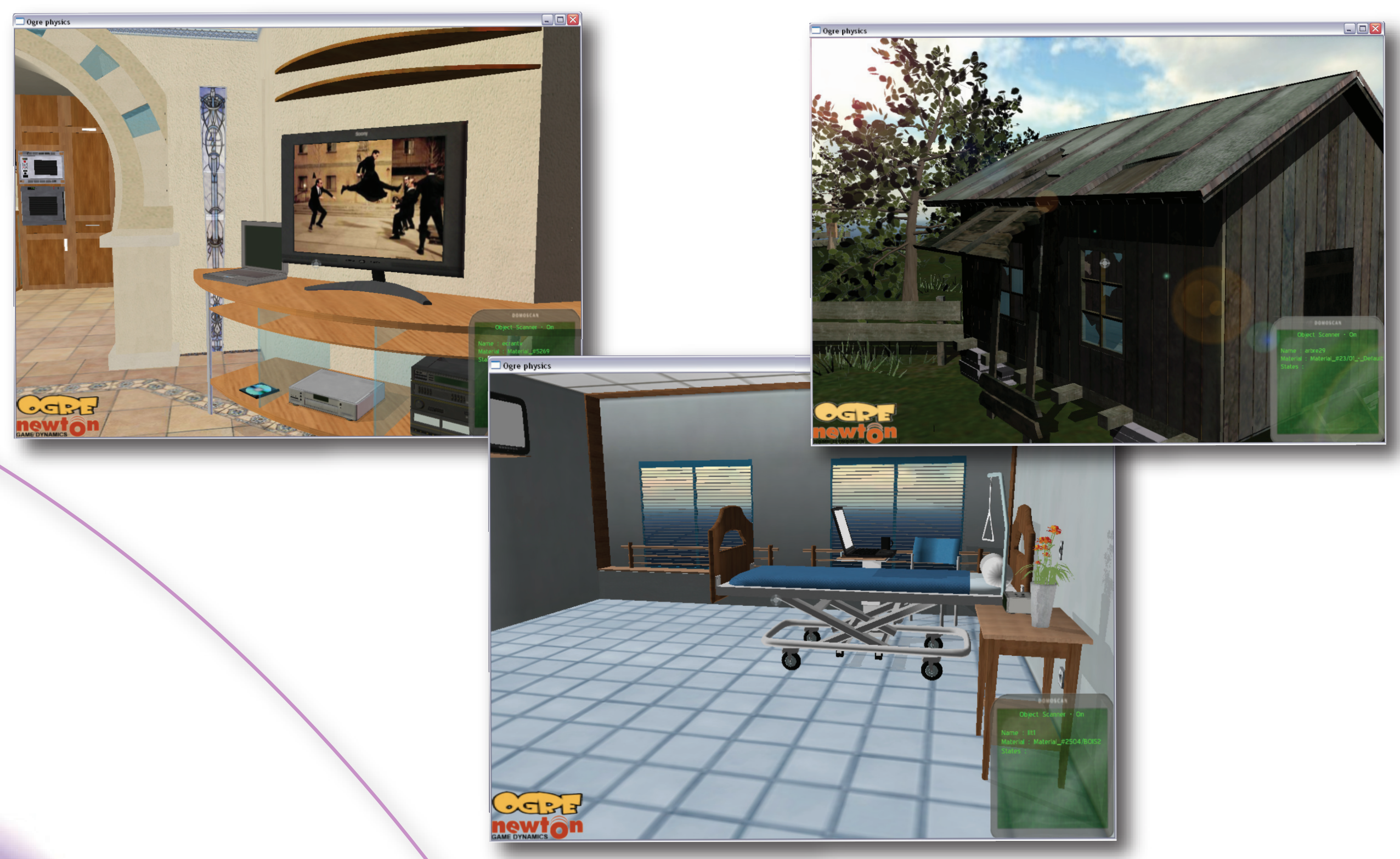


Interactions: Joystick, PDA, GSM, Wearable Computer

• Web Services

• Web Services for Virtual Devices

Simulated Environment: 3D Scene



Ubiquarium

The Ubiquarium comprises various devices and services, which can be discovered and composed at runtime. Those devices can either be virtual devices (3D scene objects in which the user is immersed), or physical devices worn by the user or present in his immediate environment. All devices of the Ubiquarium, physical or virtual, are based on Web Services interfaces which limit ad-hoc programming.

The current Ubiquarium relies on three major categories of devices:

- Physical environment of the user: wireless devices actually present in the environment such as sensors (luminance, temperature, accelerometer, ...) and actuators (remote relays).
- Worn by the user. Interaction devices: joysticks, cell phones, PDA, wearable computer.
- Simulated environment: a virtual 3D scene, virtual UPnP devices related to scene objects.

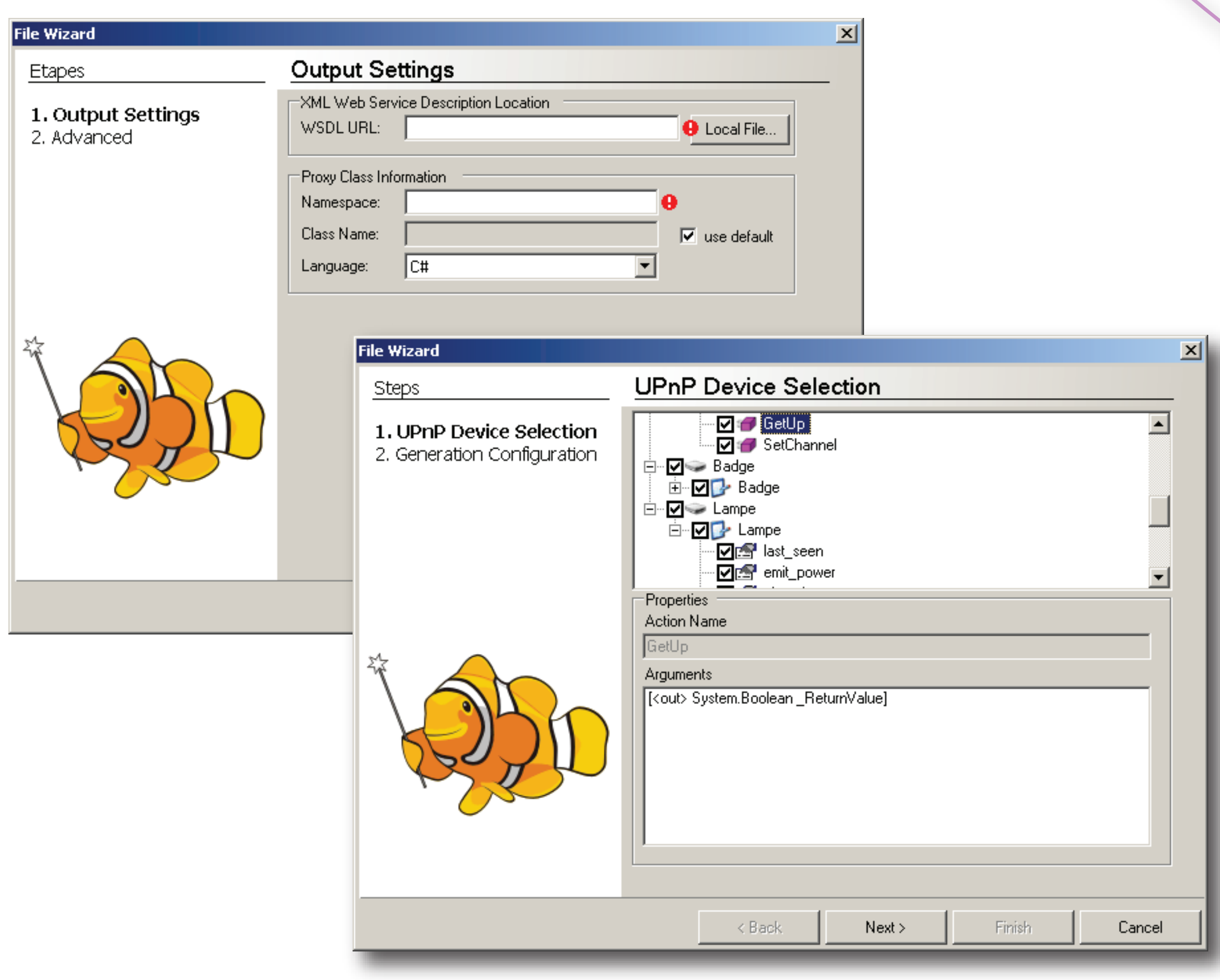
This represents an ideal framework for the evaluation of new applications for mobile and ambient computing like the usage of wearable computers. This platform is particularly well adapted to the study of the software adaptation mechanisms for context-aware mobile and ubiquitous computing applications.

WComp

The paradigm, which allows the management of the orchestration of those web-services for devices, is particularly relevant when it is based on an assembly of components, namely associated with a dynamic-composition language. According to contexts and available devices, different aspects of the application can be selected, applied and then woven to adapt the application. Wcomp is a prototyping environment for context-aware applications.

The Wcomp Architecture is organised around what we call Containers and Designers. The purpose of the Containers is to take into account system services required by components of an assembly during runtime: instantiation, destruction of software components and connectors. Designers allow configuring assemblies of components through containers. A graphical designer such as Bean4Wcomp enables to compose manually assembly of components according to a graphical representation of the event flow for instance. It is particularly adapted to the description of the application. A designer based on Aspects of Assembly (called ISL4Wcomp) allows describing interaction patterns thanks to an evolution of ISL (Interaction Specification Language). Interaction patterns are then selectable, applicable and woven and allow adapting dynamically the application to its context.

Generation of Components



File Wizard

Etapes

1. Output Settings
2. Advanced

Output Settings

XML Web Service Description Location

WSDL URL:

Proxy Class Information

Namespace:

Class Name:

☑ use default

Language: C#

File Wizard

Steps

1. UPnP Device Selection
2. Generation Configuration

UPnP Device Selection

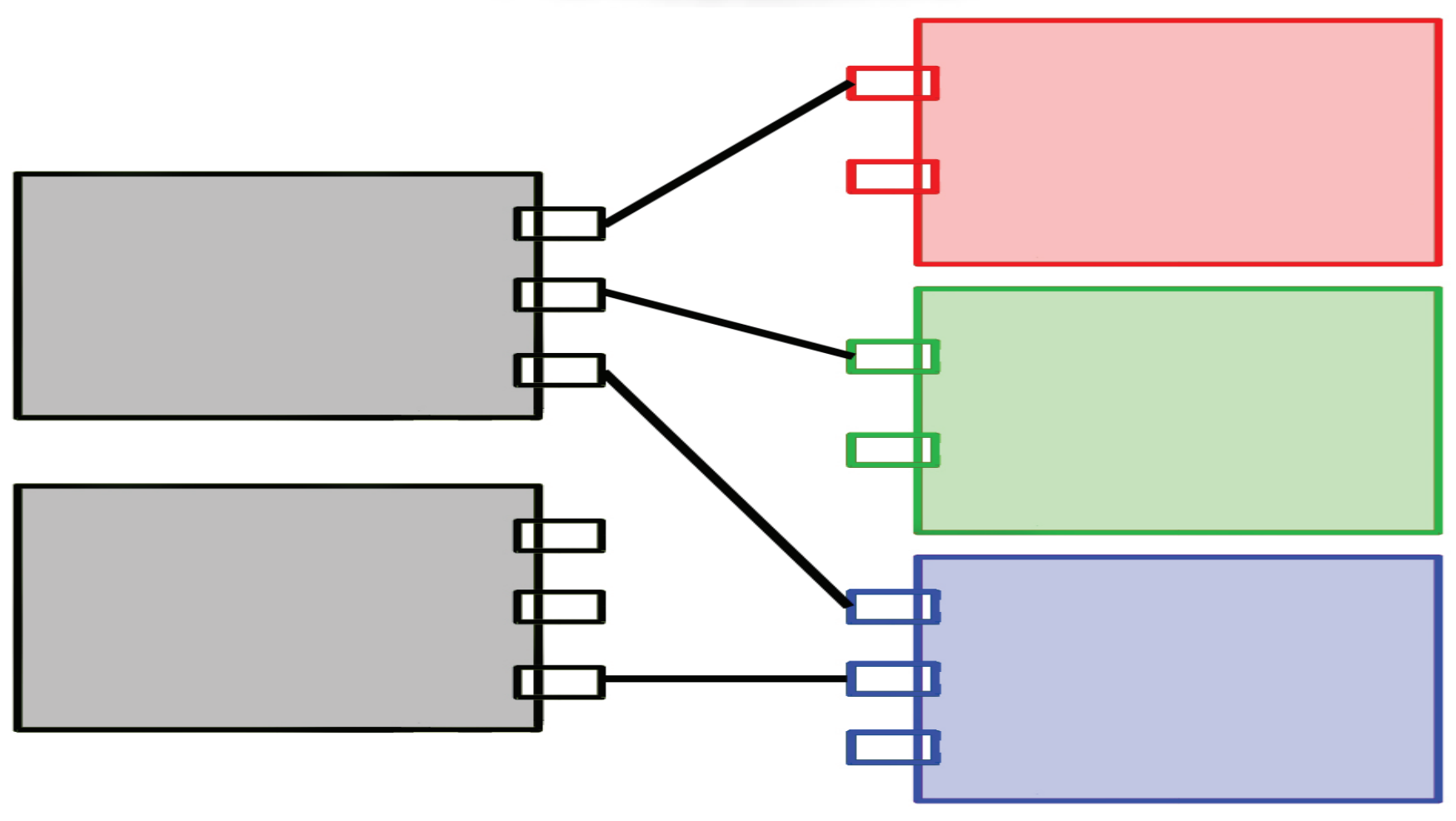
☑ GetUp
☑ SetChannel
☑ Badge
☑ Badge
☑ Lampe
☑ Lampe
☑ last_seen
☑ emit_power

Properties

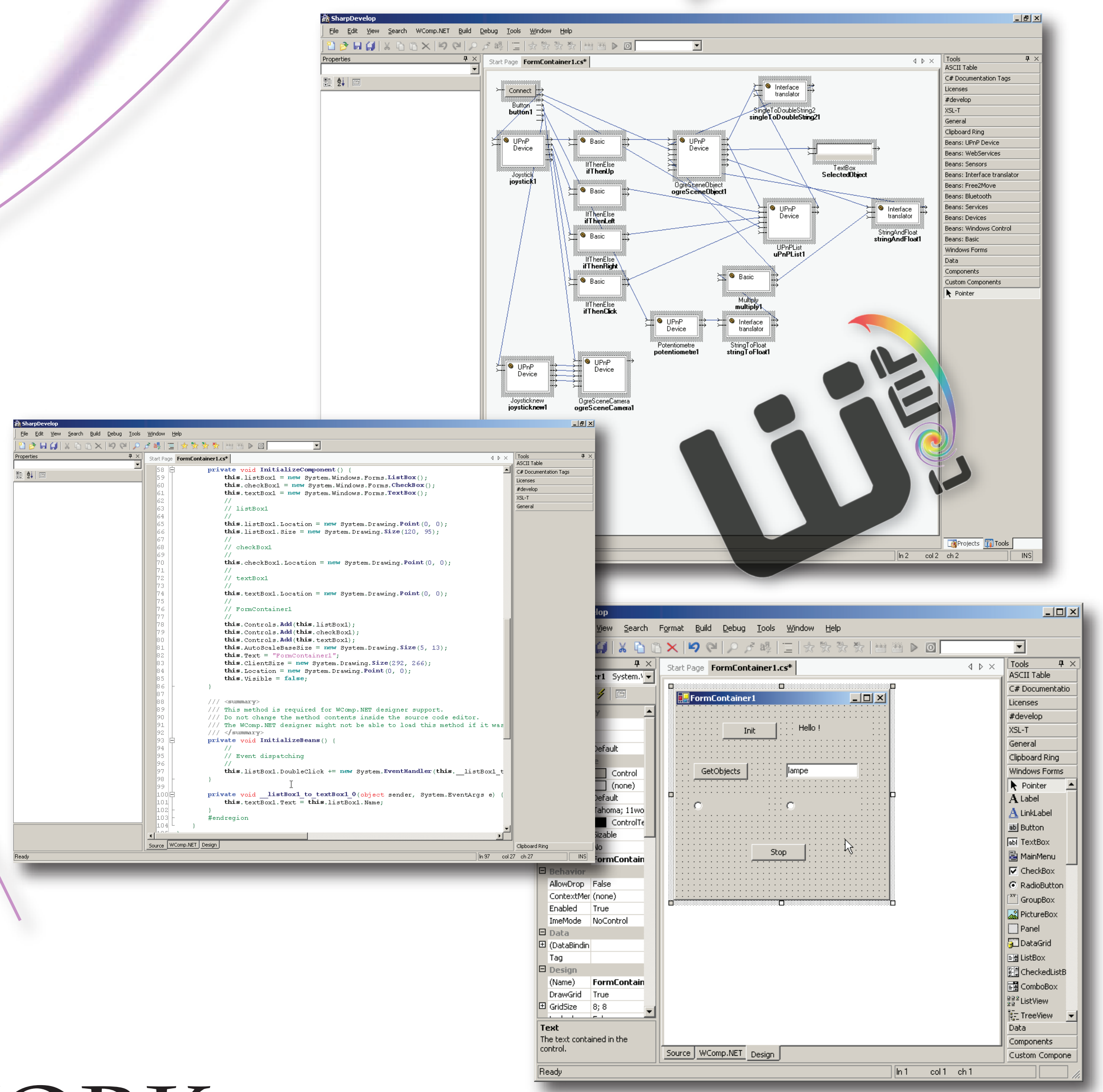
Action Name: GetUp

Arguments: [<out> System.Boolean _ReturnValue]

Containers



Designers



AN ORCHESTRATION FRAMEWORK

For any information: http://www.polytech.unice.fr/ubiquarium/

Région
Provence-Alpes-Côte d'Azur

Polytech'Nice-Sophia

i3S sophia antipolis

CNRS
CENTRE NATIONAL
DE LA RECHERCHE
SCIENTIFIQUE

Université Nice SOPHIA ANTIPOLIS

* From Latin *Ubique*, everywhere, with the suffix *rium* meaning location and structure. Hence, Ubiquarium means: "a location or a structure in which a computer is everywhere and in everything".