FP7 Project

Grant Number 288238

VENTURI platform integration

Outline

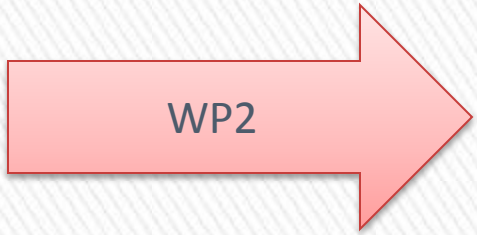
- Y1 demonstrator workflow and software stack overview
- Y2/Y3 demonstrators integration effort
- Algorithmic solutions overview
  - ST-Italy, Metaio, FBK, HHI, Inria illustrate:
  - High level picture of algorithms functionalities
  - Software environment, architecture and dependencies
- Open discussion on possible integration solutions, potential opportunities and issues
- Define actions, homework and deadlines for integration efforts

Y1 demonstrator workflow

- Use case definition lead to platform requirements
- Platform requirements ended up in platform specification
- Platform specification augmenting existing STE platform
- Main actors
  - Metaio: Augmented reality SDK provider
  - ST-Ericsson: platform provider, new requirements execution
  - E-Diam: Android application design and implementation, on top of Metaio SDK
- E-Diam application is running on top of Metaio and Android APIs

Base Platform as of D2.1

- Junaio SDK
- STE Android ICS
- Linux Kernel
- STE U8500 platform



Y1 augmented platform

- E-Diam Application
- Unifeye SDK
- STE Android ICS
- Linux Kernel
- STE U8500 platform

Y2/Y3 demonstrators

» Y2/Y3 demonstrators will integrate not only platform enhancements derived from WP2 requirements, but also WP4 and WP5 algorithms, methods and software components coming from ST-Italy, HHI, FBK, INRIA and Metaio SDK.

Sensors based context sensing

Camera based context sensing

Audio based context sensing

3D based reconstruction

3D based rendering

WP2/WP4/WP6

Y2/Y3 platforms

- E-Diam Application
- Metaio SDK
- STE Android ICS
- Linux Kernel
- STE U8500 platform

Integration at different levels !

Integration rationale

- We need to appraise, from a system point of view, all partners software components assets and come up with a consistent and realistic integration plan for Y2/Y3 prototypes.
- Target: D4.3 “WP4 outcome definitions and API specifications for inter-task / inter-WP communications” (M15)

- Each algorithm provider partner to illustrate:
  - High level overview of their algorithmic software solution
  - Detail software assets in terms of:
    - Reference platform: x86, ARM, other,…
    - Reference OS: Ubuntu/Linux, Android, iOS, Windows,…
    - Language: C, C++, Java, Objective-C, assembler,…
    - Libraries and/or frameworks dependencies: Glib, GStreamer, OpenCV, Eigen, D-Bus,… **VERY IMPORTANT**
    - Reference build system: Eclipse, autotools, Ant, make, scons,…
    - Solution “openness”: completely closed, binary only release, source release
  - High level view of software implementation architecture

Integration bootstrap

Open discussion