Software Integration for the ASTRI SST-2M prototype proposed for the Cherenkov Telescope Array

V. Conforti[1], E. Antolini[7], L.A. Antonelli[4], P. Bruno[2], A. Bulgarelli[1], R. Canestrari[3], O. Catalano[6], A. Costa[2], V. Fioretti[1], S. Gallozzi[4], F. Gianotti[1], Grillo[2], F. Lucarelli[4], G. Malaguti[1], M. Mastropietro[4], F. Russo[5], P. Sangiorgi[6], J. Schwarz[3], S. Scuderi[3], C. Tanci[7], G. Tosti[7], M. Trifoglio[1], V. Testa[4], S. Vercellone[6], F. Vitello[2], A. Zoli[1], M. Bartolini[10] and A. Orlati[10]
for the ASTRI Collaboration[8] and the CTA Consortium[9]

[1]INAF, IASF-Bologna, Via Pietro Gobetti 101, 40129 Bologna, Italy
[2]INAF Osservatorio Astronomico di Catania, Via S. Sofia 78, Catania, Italy
[3]INAF – Osservatorio Astronomico di Brera, Via Bianchi 46, Merate, Italy
[4]INAF – Osservatorio Astronomico di Roma, Via di Frascati 33, Monteporzio Catone (Roma), Italy
[5]INAF – Osservatorio Astronomico di Torino, Via Osservatorio 20, Pino Torinese, Italy
[6]INAF IASF-Palermo, Via Ugo La Malfa 153, 90146 Palermo, Italy
[7]Università di Perugia, Via A. Pascoli, Perugia, Italy
[8]http://www.brera.inaf.it/astri, [9]http://www.cta-observatory.org
[10]INAF IRA. Via Fiorentina 3513, 40059 Medicina (BO), Italy

INTRODUCTION

INAF is leading the ASTRI project within the Cherenkov Telescope Array (CTA), for the realization of an end-to-end small sized telescope prototype, and a mini-array of at least nine ASTRI telescopes proposed to be placed at the CTA southern site. The ASTRI Mini Array Software System (MASS) is the software system that oversees the telescope monitoring, control, data analysis and archive, as well as data access and dissemination at any level, including user or hardware. The ASTRI SST-2M prototype is installed in Italy at the INAF “M.G. Fracastoro” observing station located at Serra La Nave on mount Etna, Sicily. It is being verified and tested with the engineering software released in beta version. Since we are adopting an iterative incremental approach for the development of the software, it is growing and changing rapidly. The software integration team’s goal is to integrate more software components, each of which functions properly in standalone tests, into the larger ASTRI software system that fulfils the system requirements. An additional goal is to support the maintenance activities such as debugging and the implementation of new functionalities. The drivers for the software integration activities are the ASTRI software requirements, both in text and use case form, the software and the component interfaces. Before starting these activities we have defined the integration model and the development rules.

INTEGRATION MODEL

In information technology, the systems integration is the process of linking together different computing systems and software applications physically or functionally, to act as a coordinated whole. The ASTRI Collaboration and CTA Consortium selected the OPC-UA (Unified Architecture) to connect the software to the hardware devices, and ACS (Alma Common Software) to interface the software components to each other within the application layer (Figure 1). We plan three levels of integration:

1. Lowest level to test the interfaces between each hardware device and its related software;
2. Middle level to test the interfaces between a software component and its neighbours (incoming and out coming calls);
3. Top level to test the system as a whole by executing the scenarios detailed in the use cases.

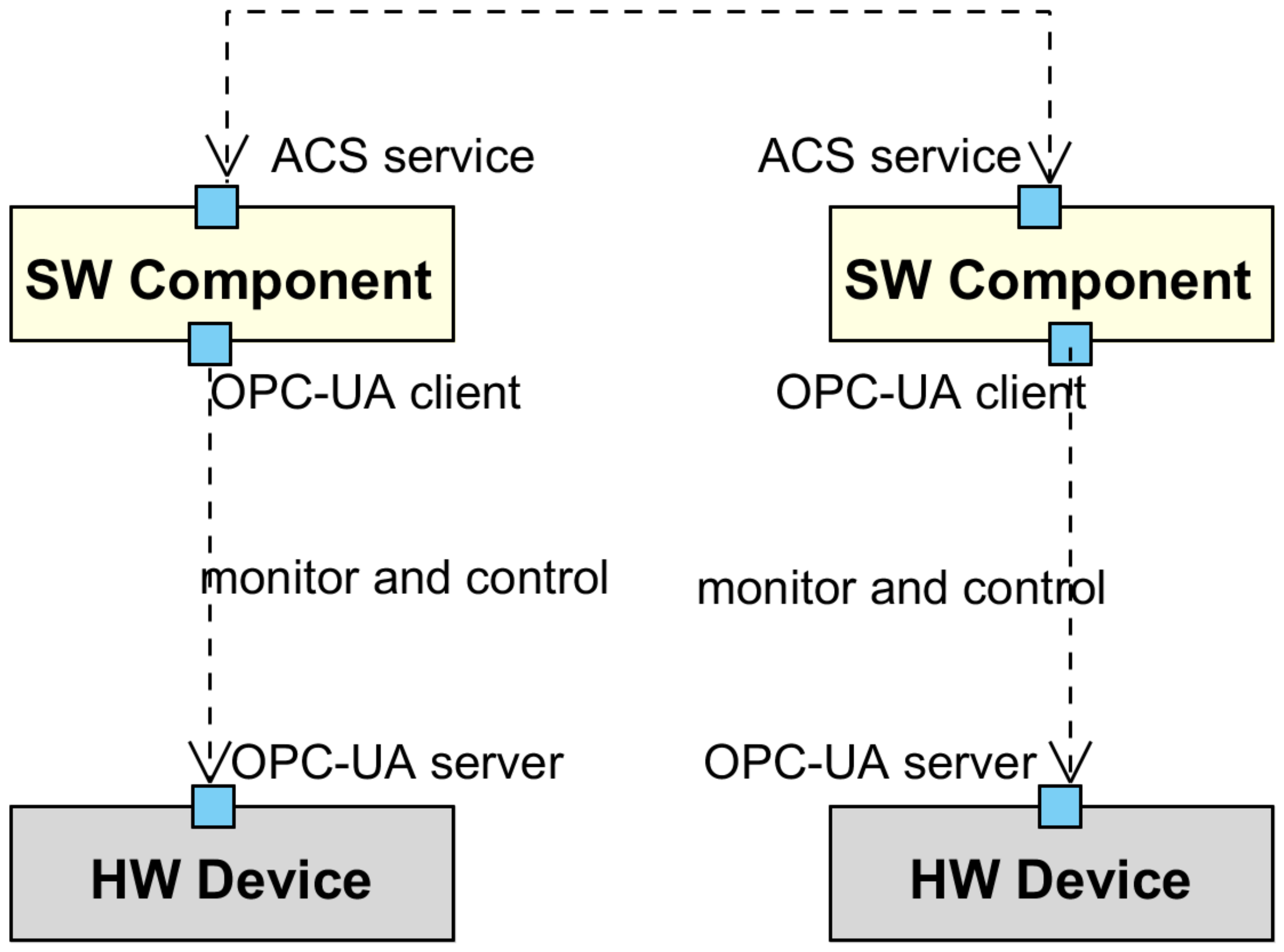

Figure 1

In order to support the software development and integration activities we make use of a series of tools: Git, Jenkins, Docker, virtual machines, test bed, and the documentation templates.

DEVELOPMENT RULES

The integration activities require specific development rules to support the integration processes:

Development Environment

- It is strongly recommended to use the virtual machine for the development which provides a standardized environment.

Lowest Level Integration

- It is mandatory to perform the lowest level integration (for components which interface hardware devices) and middle level integration (exploiting Docker machine).

Component Release

- The software component must be tested and released with its user manual and setup instruction. It is mandatory to use the Git repository for the releasing of software.

Top Level Integration

- It is mandatory to perform the top level integration (exploiting the test bed) before any major release of the software, and before the deployment in the production environment.

Continuous Integration

- Continuous integration, consisting of a full build and unit tests execution, is done via Jenkins.

CONCLUSIONS

The method and the tools to perform the software integration activities for the ASTRI SST-2M prototype are starting to be used as of this writing. We will refine the procedures as our early experience dictates. The lessons learned with these activities will provide valuable input for the production phase of the CTA software.

***Acknowledgements**: This work is supported by the Italian Ministry of Education, University, and Research (MIUR) with funds specifically assigned to the Italian National Institute of Astrophysics (INAF) for the Cherenkov Telescope Array (CTA), and by the Italian Ministry of Economic Development (MISE) within the “Astronomia Industriale” program. We acknowledge support from the Brazilian Funding Agency FAPESP (Grant 2013/10559-5) and from the South African Department of Science and Technology through Funding Agreement 0227/2014 for the South African Gamma-Ray Astronomy Programme. We gratefully acknowledge support from the agencies and organizations listed under Funding Agencies at this website: http://www.cta-observatory.org/.*