Towards Vendor-Agnostic IT-System of IT-Systems with the CEDE Platform

A. Luis Osório(✉)

ISEL - Instituto Superior de Engenharia de Lisboa,
Instituto Politécnico de Lisboa, and POLITEC&ID, Lisbon, Portugal
lo@isel.ipl.pt

Abstract. The increasing capabilities and quality of information technology has thrown integration endeavour to a top priority (total integration trend). However, the lack of time to mature consensus and the market dynamics, has led to technological dependencies (vendor lock-in). Over the past few years, a variety of new concepts, paradigms, methodologies and technologies have contributed to unique information technology (IT) solutions. One main research question is how to establish an open development landscape for the emergent complex integrated IT system of systems under vendor agnostic models (openness). The Collaborative Enterprise Development Environment (CEDE) is a research effort towards a unified development culture (based on conformity certifications). The strategy is complementary to existing standardizations processes and is based on promoting an open (unified) collaborative development environment.

Keywords: Collaborative networks · System of systems · Modularity framework · Collaborative development

1 Introduction

As end-users of information and technology systems (IT-systems) public and private organizations, are facing a crescent problem on how to govern their IT assets under a sustainable model, i.e., how can they ensure that the life cycle management of their assets is under a market competition model? This establishes the vendor-agnostic goal, making a technology landscape potentially sourced from competing suppliers without substitution constraints. At present such substitution is not, in most of the cases, easy or even possible. The IT-systems, potentially needed customizations/configuration efforts after acquisition, in most of the cases, involving considerable costs. These efforts strongly depend on the specific culture of the supplier of the acquired IT-System. IT-systems like enterprise resource planning (ERP), customer relationship management (CRM), supply chain management (SCM), are based on different development strategies (architecture and technologies, and parametrization issues) depending on its supplier. Important standardization efforts, like the business process model and notation (BPMN) from Object Management Group (OMG) are not an answer to avoid the need for specificities associated to technology bindings, as one of the problems that makes a

H. Afsarmanesh et al. (Eds.): PRO-VE 2016, IFIP AICT 480, pp. 494–505, 2016.
DOI: 10.1007/978-3-319-45390-3_42

substitution of a system a complex and costly process. No standards (complete) approaches are known as addressing the current handcrafted configuration efforts, by establishing specific bindings between models and implementation structures.

This problem has been addressed by both the research community and the industry in a number of projects and standardization efforts targeted to (re)think the IT landscape. The FIWARE European initiative is an example of such an effort to promote the substitutability of the (computational) responsibility of IT-systems by assuring that the substitute completely replaces the existing system without the need for any additional configuration/adaptation effort. The FIWARE initiative establishes the Generic Enabler and the reference implementation (GEri) concepts as an external modularity strategy. The GE abstracts a specialized computational responsibility through a set of interfaces for the cooperation with other IT-systems. This means that a FIWARE enabled product shall be conform a GEri used in the validation process (conformity certification) [18]. This initiative assumes however that the approach needs further efforts on defining the complete standards suite able to make the vision an effective contribution for a multi-supplier framework. The IT4IT is another effort from the well-recognized normalization body OpenGroup aiming at establishing "A Reference Architecture for Managing the Business of IT" under a main motivation of working across the recognized silos and the need for novel value chain where the substitutability is facilitated [12]. However, even if vendor-agnostic is declared as an important concern, in IT4IT reference architecture it is assumed that beyond the level 3 (levels 4 and 5) the approaches are vendor specific, Fig. 1.

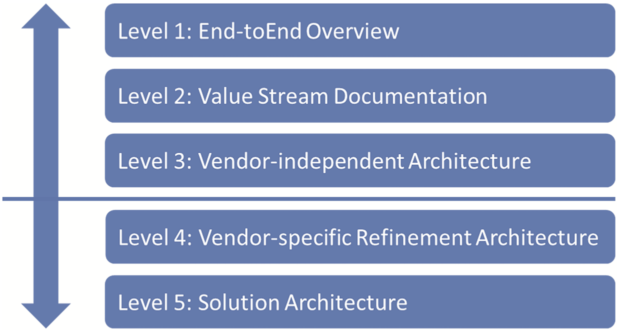

Level 1: End-toEnd Overview
Level 2: Value Stream Documentation
Level 3: Vendor-independent Architecture
Level 4: Vendor-specific Refinement Architecture
Level 5: Solution Architecture

Fig. 1. IT4IT reference architecture levels [12]

This and other efforts demonstrate a crescent concern about the lock-in problem. They represent strategies to reduce cross border silos and the related risks, and added costs difficult to moderate (coordinate). At least for critical systems, there is a trend to adopt monolithic solutions under a unique responsibility from a single supplier, as a strategy to guarantee minimal risks.

This paper discusses the CEDE research as a contribution for the vendor-agnostic of IT-system of IT-systems, under an open research challenge, considering both open modularity, and a unified development culture. Both approaches are related to previous research work and productized IT solutions, adopted by the industry.

2 Open Complex System of Systems Research

The development of complex systems is a long discussed research and industry community concern. In 1972 Liskov [8] discussed reliability and complexity associated to the software developments. The concern was not at that time the vendor-agnostic but rather the complexity of developing reliable software systems. The introduction of the term complexity refers to, in the one hand "*…there are many system states in such a system, and it is difficult to organize the program logic to handle all states correctly*" and on the other hand "*… the efforts of many individuals must be coordinated in order to build the system*". It is quite interesting that more than four decades later we are discussing more or less the same problem. While nowadays the complexity might have increased, and the state of the knowledge about computer science and engineering and the available resources have evolved under paramount dimensions, we still are on difficulties to offer reliability for the new integrated and distributed system of systems. Or at least, such reliability is difficult to guarantee in a multi-supplier environment where suppliers holding different process and technology culture, collaborate to the life cycle of integrated IT-system of IT-systems (SoS).

The panoply of languages, development frameworks and paradigms are one main reason for the need for common methodologies, tools and resources able to cope with the required holistic (systemic) approaches. In fact, if considering the two technology specific main lines, the leaded by Microsoft (.NET, WCF, C#, etc.) and the Java and open source world (RMI, OSGi, Java, etc.) there is a consensus about the advantage (less risks) to adopt the unified Microsoft culture. There are a number of potential suppliers offering development services certified under this development and execution environments culture (proprietary). For what is mentioned as the open source java world (involving a diversity of many other paradigms), the situation is different; it is of high risk the adoption of a single (unique) development and process cultures. For the open source world such diversity of paradigms induce high risks when there is a need to change the supplier. There is a high probability that the new subcontracted company needs extra time to understand the assets and the potential founded need to move parts to its own development culture. This has led to situations where reutilization of existing assets is not possible or at least, is not of the interests of the new subcontracted development culture.

This situation can be in a large extent reduced if adopting certified competencies on proprietary technology landscapes (Microsoft, SAP, TIBCO, Oracle, Cisco, and many other proprietary cultures). When the strategy is to adopt open source and open specifications not leaded by a unique technology supplier it is more difficult to find competing companies able to support and evolve assets in such open worlds.

There are a number of integrated development environments (Eclipse, NetBeans, IntelliJ, among other), code generation and dependencies management (Maven, Ivy, Grape, Gradle, Buildr, STB, Leiningen, and more will appears with some justified advantage), issues and project management tools (Redmine, Bugzilla, Mantis, Trac, ProjectLibre, LibrePlan, OpenProject, MyCollab, Odoo, from many other), and well known proprietary tools like the suite offered by Atlassian. This diversity makes the potential advantages from adopting open crowdsourcing dynamics a risk to generate vendor dependencies, potentially worst that those associated to proprietary cultures. Furthermore, for large end-users the contracting of innovative start-ups is associated to a potential risk of adopting fashion technologies, development paradigms and methodologies. Such diversity makes difficult to construct consistent integration strategies for complex IT-systems, considering that these start-up, as potential competing suppliers, tend to adopt their own development culture.

Therefore, as standards are not complete to establish such a required open common culture for the development and life cycle management of complex integrated system of systems, there is a need to establish a common informatics engineering culture. One informatics engineering culture, means the establishment of standard technology, procedures, methodologies, tools, and a modularity framework, making specific development easily managed by competing suppliers. Such a convergence is expected to be accelerated and pulled by the end-users, driven by their need for competitive IT supplying markets, in particular governments. Wherever possible, the convergence shall be based on standard products, at least for the specialized IT-system infrastructure level systems (Radars, Road Side Units, things, controllers, etc.). However for higher level back-office and enterprise IT department systems, the efforts to converge to standards are commonly not enough to completely answer to evolving requirements. This has been motivating the development of common platforms like the COMPASS project that challenged a Collaborative Development Environment as an alternative to the existing Integrated Development Environments (IDE) [2]. The proposed COMPASS tool set is based on a triangular approach based on the Systems Modelling Language (SysML) a dialect of the Unified Modelling Language (UML) generating a COMPASS Modelling Language (CML) based on the Artisan product, the Java based COMPASS Overture toll [7] and the runtime tester (RT-Tester) for tests automation tasks. While arguing that is a formal contribution to open SoS, the adoption of the proprietary Artisan Studio from Atego raises the question about the openness of the approach. The modular open systems approach (MOSA) for the military industry, a program of the US department of defence (DoD), is grounded on five principles (i) establishment of an enabling environment, (ii) employ modular design, (iii) designate key interfaces, (iv) use open standards, and (v) certify conformance, as a strategy for a unified contractual framework for an open life cycle management of system of systems (complex warfighting systems) [15].

More recent theoretical works about component substitutability considers a dynamic reconfigurations of components evolving dynamically by adjusting capabilities on answering requirements changes. The adjustments are based on software component's level and structured on primitive operations like, instantiation, destruction, addition, removal, binding, unbinding, starting, stopping, and parameter's adaptation [6]. Nevertheless, while important for a scientific foundation of system of systems (or at a lower level, component of components or composites), a practical application is only viable if a unified culture is developed to accommodate such diversity of contributions for the development of reliable complex IT-system of IT-systems.

The System of systems and in particular IT-system of IT-systems concept, has been adopted along the last two or three decades as an idea of structuring the growing complexity. The Greek origin for system means a whole composed of many parts and is adopted for social, mathematical, healthcare, transport, logistics and other systems [5].

It is related to a strategy to get some form or cohesion through the composition of a diversity of parts (subsystems) coordinated for some value production.

3 The CEDE Strategy as Part of the MDEOS Program

The research to achieve a vendor-agnostic approach leads to the definition of Open System of systems, in a slightly different form but similar in objectives to the well-known Open System Interconnection (OSI) to the network/communication level. In fact, the most challenging goal is to achieve a systemic IT approach, based on a composite of IT-systems where for each subsystem the principle of substitutability shall apply. To better clarify the concept lets define IT-system of IT-system as a set S of IT-systems S = {S_1, S_2, …, S_n} where for each S_i there exist at least two potential suppliers guaranteeing a plug-and-play substitution. The definition of open IT-system of IT-systems (for simplicity SoS) diverge from what industry usually presents as open systems, which is centred on its promptness to integrate with any existing or legated system. In fact, any IT-system is (or should be) developed under architectural concerns on how to decompose the system into subsystems, guided by an internal reutilization strategy. A complex IT-system product is organized following such a modularity strategy under the culture of the developer company. This application of engineering best practice is classified as internal modularity, considering that such division of responsibilities is internal to the developer company. Two system architects will arrive, for sure, to different division of responsibilities (modularity framework) unless a same reference architecture is followed. In the open IT-system of IT-system formulation, each subsystem should be considered under an external modularity, meaning that it implements open (standard/public domain) interfaces. The design of such a SoS under external modularity requires the architecture derives from a reference architecture and is conform to the open specifications. The research challenge is in fact to establish an equivalent open IT-system where the end-user is more "owner", and has the governance control of the It-system of IT-systems, Fig. 2.

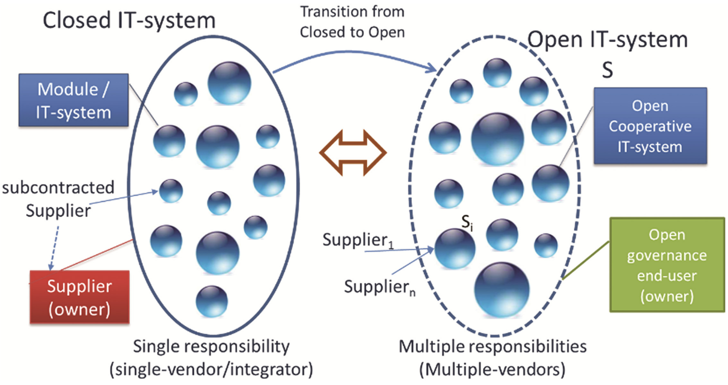

Closed IT-system
Transition from Closed to Open
Open IT-system S
Module / IT-system
subcontracted Supplier
Supplier (owner)
Open Cooperative IT-system
Si
Supplier1
Suppliern
Open governance end-user (owner)
Single responsibility (single-vendor/integrator)
Multiple responsibilities (Multiple-vendors)

Fig. 2. Open IT-system of IT-systems

The approach to this endeavour starts from a more general strategic vision to address the upcoming complex IT-systems where the today dependencies are not acceptable, considering the dependencies they impose to the innovation processes (lock to unique suppliers). The Model Driven Engineering Open Systems (MDEOS) initiative frames the CEDE platform and a previous research, the Cooperation Enabled System (CES) [13]. The CES modularity framework, aiming at establish an external modularity framework for SoS, was strategically postponed by the difficulties to operationalize the proposed model. The reference implementation started to get too complex and the number and diversity of adopted open source contributions has generated a complex implementation difficult to validate and to be accepted by the market as an open reference modularity framework. The MDEOS program, Fig. 3, was established on the assumption there exists a gap between processes and technology that needs to be shortened and, if possible, removed through the establishment of a suite of coordinated model driven tools.

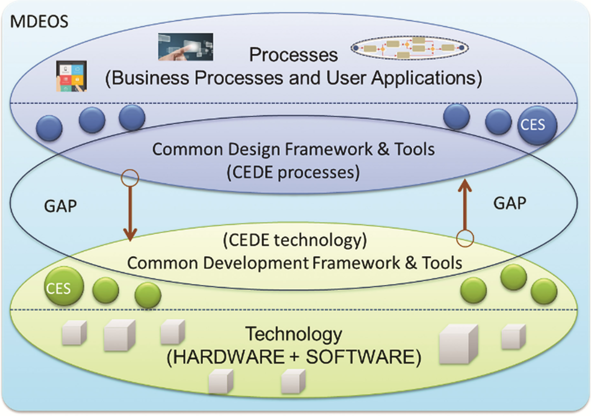

MDEOS
Processes
(Business Processes and User Applications)
CES
Common Design Framework & Tools
(CEDE processes)
GAP
GAP
(CEDE technology)
Common Development Framework & Tools
CES
Technology
(HARDWARE + SOFTWARE)

Fig. 3. Open IT-system of IT-systems

The gap between processes and technology has long been recognized as a major challenge for information intensive companies [9]. The model driven engineering (MDE) and model driven development (MDD) research, motivated in part from the Model Driven architecture (MDA) principles as formulated by OMG [11], have been contributing to reduce the gap by helping to automate the bridge between de platform independent models (PIM) and the platform specific ones (PIM). However, in spite of the number of contributions related to the MDA challenge, the dependencies from specific organization's models and meta-models and technology approaches, remains as a main obstacle for the establishment of open system of systems and this way contribute to reduce the gap between processes and IT technology.

Among the research towards the establishment of accurate models, a valuable contribution to reduce the gap between models and implementations is proposed by the Object-Process Methodology (OPM) [17]. Considering its simplicity and the mechanisms provided to describe complex system of systems, it is planned to include this modelling tool into the CEDE paradigms portfolio.

Both CEDE and CES are research contributions, as convergence efforts for the reduction of such a gap, by promoting, in the one hand, a unified development environment (CEDE) and, on the other hand, by establishing a high abstraction level to encapsulate computational responsibilities. This CES research aims at establish a complete and consistent vendor-agnostic suite of capabilities able to commit to the substitutability principle (open modularity framework).

The CEDE implementation is under development and validation, and in this phase it is focused on the selection of methodologies, technologies and tools by establishing this way a common standard development framework for the technology perspective. A more long term objective is to extend the CEDE platform to contribute for the other line of the gap with an enhanced version offering a common design framework and tools for the processes domain (as a strategy to reduce it). The idea is to offer two main groups of specialized tools, one for technology developments and the other for accessing process experts to perform the required agile adaptation to requirements change.

4 CEDE Framework Development and Validation

The initial version of the CEDE platform considers the following mandatory options: (i) the projects are structured based on the Apache Maven framework supported by Git version control and Nexus repository management; (ii) the Eclipse integrated development environment unifies the tools, namely a tight coordination with the Maven mechanisms (through M2E plugin); (iii) The java language and the OSGi specification were selected as mandatory approaches; (iv) the Redmine[1] issues and project management tool was selected as a main contributor to establish a development collaborative environment, eventually enhanced with specialized plug-ins; and (v) a federated authentication and authoring infrastructure based on LDAP directory unifies access control. The selection of technology paradigms and tools were guided by openness and a founded recognition by both the industry and research community. For the potential question, about why adopting OSGi and not other modularity framework, the reason is associated to its intrinsic potential and recognition based on a crescent adoption by the industry. Even if the OSGi specification is focused on Java, there are research works evaluating the possibility to bind this framework to other programming languages, e.g. C#/.NET as discussed in [3]. Following similar founded decisions, the construction of the CEDE platform is based on a Linux server (Ubuntu), and it is planned to be deployed from a reference implementation able to speed-up its adoption. The CEDE servers might be running on premises or on the cloud, as a resource dedicated to manage and coordinate collaborative developments for complex large scale IT-systems, Fig. 4.

[1] Redmine - http://www.redmine.org/.

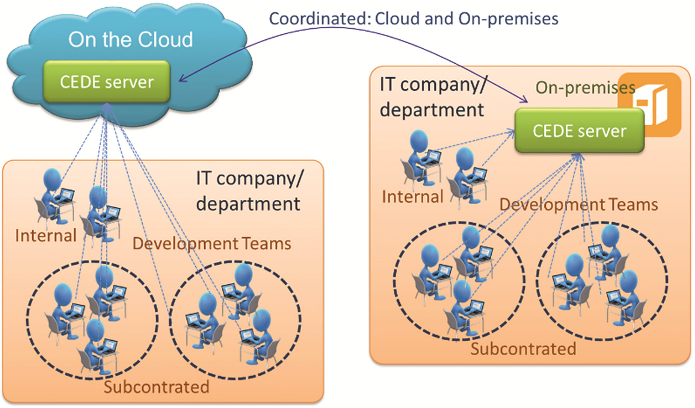


Fig. 4. The new PCS IT-system wraps the legated PSW/JUP

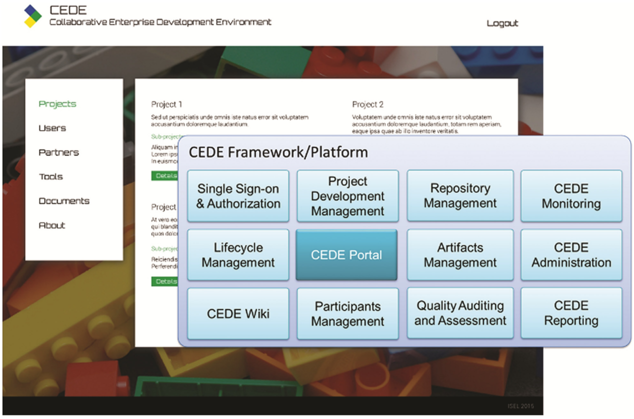
The selection of specific frameworks or computing paradigms is difficult to decide when there are more than one, with strong valuable arguments. One extreme example about such decision difficulties is the adoption by the OSGi standardization of two competing paradigms: i) the Blueprint Container (osgi.121, BC) and the Declarative Services (osgi.112, DS) as frameworks realizing the dependency injection and inversion of control patterns (DI/IoC). While being argued for the structuration of large scale object-oriented applications [1], the OSGi standard seems however to be conditioned by industry investments and even if the Declarative Services (DS) paradigm is getting momentum, any product that is developed on Blueprint Container is also conform to the standard. This is an example of the extreme difficulties to converge to consensus on deciding for single approaches when many options have their own arguments to be the selection one.

The CEDE platform is therefore a strategy for the development of open complex large scale IT-system of IT-systems grounded on an enhanced collaborative ecosystem. It is grounded on simple basic principles: (i) an open unified culture established by the suite of adopted concepts, implementations, technologies, methodologies, tools, and techniques; (ii) a concretization through a CEDE server with a reference implementation from where platform instances can be derived; and (iii) a competencies certification process both for developers and for companies. One main objective is to establish a vender-neutral IT-system development culture and this way reduce the tacit knowledge common to software development processes. As a knowledge intensive activity, many strategies have been researched namely the adoption of a robust transitive memory systems (TMS) to support the knowledge sharing among software development teams [16].

Furthermore, the tacit knowledge associated to the software development process is not only a problem for IT product development companies but also for the end-users. The lack of standard off-the-shelf IT-systems results on the need for specific developments, as the current and most usual strategy to make capabilities to fully match the requirements. The end-user companies, public organizations and authorities have many difficulties to establish competitive tenders for IT capabilities, based on the discussed risks of dependencies they might induce. The public tenders usually specify what is required but not how the capabilities should be organized, making this way budgeting very difficult and the resulting system dependent from the culture of the winning supplier. These aspects are main reasons to pretermit start-ups and small and medium companies (SME) in favour of less risky large well known IT suppliers with their own culture.

Therefore, beyond the pragmatic suite of principles the CEDE platform establishes an open IT development culture. It aims at contribute to an enhanced open competitive market, making this way easier for start-ups to prove about the capabilities of their products. The risks are reduced based on the substitutability principle resulting from the adoption of the CEDE IT development unified culture. A first approach to the CEDE platform is being validated by the migration of the ECoNet [14] and Horus projects. The Horus project developed the Petrol Distributed Open Service Bus (PDOSBus) as a suite of cooperative services to manage post-payments in a gas-station forecourt. The Horus IT-system obtains the vehicle identification through automatic licence plate recognition (ALPR) and registers and manage payment incidents when the vehicles leave the forecourt without paying. When a vehicle with a payment incident is positioned to feel the gas-tank and a person removes the nozzle, the point-of-sale subsystem (POS) asks a specialized Horus service for any payment incident for that vehicle. If there is a pending payment the POS operator is presented with a special warning symbol making him to initiate the suggestion for a pre-payment procedure to that vehicle. While not a too complex IT-system of IT-systems the participation of two start-ups (Exploitsys and Makewise), a large IT supplier for the POS subsystem (Gasodata/Dresser Wayne) and two large end-users companies (Galp and BP) makes the case well suited for an initial validation.

The CEDE platform while based on a suite of well-known tools, as above discussed, is planned to have a main entrance portal from where collaborating companies and individual developers can access to perform their contributions. The main modules of CEDE



Sed ut perspiciatis unde omnis iste natus error sit voluptatem accusantium doloremque laudantium.

Voluptatem unde omnis iste natus error sit voluptatem accusantium doloremque laudantium, totam rem aperiam, eaque ipsa quae ab illo inventore veritatis.

Fig. 5. A first approach to the CEDE unified collaborative development environment

platform include: (i) a single sign-on and authorization module, based on LDAP and a federated identity management); (ii) project and issues management (based on Redmine); (iii) a repository management (based on Nexus) and Git versioning system; and other governance, documentation, and management tools, Fig. 5.

As a turnaround approach to answer the lack of an effective multi-supplier (open) modularity framework, the CEDE platform aims at establish an open collaborative space. Such unified collaborative development space aims to establish a unified development environment based on a pre-established suite of standards mechanisms, technologies, patterns, and theories able to promote an enhanced competitive ecosystem for the life cycle management of open large scale complex IT-system of IT-systems. The CEDE research is aligned to the Software Engineering Method And Technology (SEMAT) concerns about the lack of solid scientific foundations for the new complex IT-system (of systems) development [4, 10]. While the SEMAT concerns are focused on software development, and the challenged program is of paramount importance, the CEDE in particular and the MDEOS program in general advocates for an IT system thinking at a different level, complementary to the software engineering layer. It is difficult to maintain a robust modularity bases when process requirements are mapping directly to software development decisions.

The CEDE platform was motivated by the difficulty large IT development companies have to manage the subcontracted partners to answer specific development projects. It is not easy for the development manager to establish reutilization strategies when the subcontracted companies hold different process and development cultures. While large companies impose their own development culture to networked partners, the model while common, has the drawback to establish proprietary solutions. Like it happened in other industries, e.g. the automotive industry, it is expected a convergence for a hierarchical market structure where larger companies positioned themselves as integrators responsible for IT-system of IT-systems based on a network of smaller specialized companies as suppliers of IT-systems to incorporate composites under the responsibility of the larger IT integrator company. The CEDE research aims at contribute for an open approach equivalent to the already existing based on proprietary cultures.

While partial contributions were identified (FIWARE, MDA, SEMAT, from other), no other known approach was found to answer similar objectives. A huge number of open source initiatives are following partial approaches to get visibility and the credibility from the market (e.g. the case of the openHAB - a vendor and technology agnostic open source automation software for your home). While many are getting success, most of them are not able to get momentum, a risk that might be reduced if adopting an open IT-systems development culture.

5 Conclusions and Further Research

The Collaborative Enterprise Development Environment (CEDE) research, while part of the more ambitious Model Driven Open Systems (MDEOS) program is being founded on the experiences and drawback from previous research (CES, ECoNet, Horus, ITSIBus, and PDOSBus) sponsored by large public and private companies (Brisa, Galp, BP, APL and APDL - Administrations of Port of Lisbon and of Leixões) and governmental agencies (INIR/SIEV, ANSR). It is founded on the need for an open development culture, at both IT systems and software development levels and also at business process application domain (functional areas). The research question, how to develop open multi-supplier IT-system of Systems able to cope to the crescent complexity associated to total integration of intra and inter organization (collaborative) processes was partially answered by the proposal of the CEDE strategy. The concept of internal and external modularity were defined, being that external modularity plays a key role on the development of open SoS. By committing to external modularity principles, it means an IT-system is able to be substituted by a competitor, i.e., it is prepared for the cooperation and to be dynamically plugged to a composite making a system of systems (SoS). The unification of the development culture is also a key approach as a contribution for an ecosystem made of pluggable IT-systems and developments under specific requirements.

One main open research question is how the collaborative networks can contribute to the structuration of the underlying complex interactions among companies participating in the development of complex large scale integrated IT-systems. The collaborative developments are expected to play a major role by promoting value creation networks where companies of different sizes, from start-up to large suppliers collaborate on the development of complex critical IT-systems.

Acknowledgements. This work has been partially supported by Administration of the Port of Lisbon and Leixões through the MIELE project, Brisa Innovation and Technology, through a yearly research grant, Galpgest and BP through the Horus project and ANSR (National Road Security Authority) through the SINCRO project. The reference implementations has been supported by a group of fellowships, Paula Graça, Inês Guilherme, Paulo Borges, Victor Camargo, and Calmenelias Freitas. The participation of Exploitsys and Makewise IT companies is of paramount importance.

References

1. Alshara, Z., Seriai, A.-D., Tibermacine, C., Bouziane, H.L., Dony, C., Shatnawi, A.: Migrating large object-oriented applications into component-based ones: instantiation and inheritance transformation. In: Proceedings of the 2015 ACM SIGPLAN International Conference on Generative Programming: Concepts and Experiences, GPCE 2015, pp 55–64. ACM, New York (2015)
2. Coleman, J.W., Malmos, A.K., Larsen, P.G., Peleska, J., Hains, R., Andrews, Z., Payne, R., Foster, S., Miyazawa, A., Bertolini, C., Didierk, A.: Compass tool vision for a system of systems collaborative development environment. In: 2012 7th International Conference on System of Systems Engineering (SoSE), pp. 451–456, July 2012
3. Escoffier, C., Donsez, D., Hall, R.S.: Developing an osgi-like service platform for .net. In: 3rd IEEE Consumer Communications and Networking Conference, CCNC 2006, vol. 1, pp 213–217, January 2006

4. B Ivar and Richard. The semat-software engineering method and theory-vision satement (2009). http://semat.org/documents/20181/27952/SEMAT-vision.pdf/16059a36-a0ba-4405-b883-4a11a2131cea
5. Karcanias, N., Hessami, A.G.: Complexity and the notion of system of systems: part (i): general systems and complexity. In: 2010 World Automation Congress (WAC), pp. 1–7, September 2010
6. Lanoix, A., Kouchnarenko, O.: Component substitution through dynamic reconfigurations. In: Proceedings of 11th International Workshop on Formal Engineering Approaches to Software Components and Architectures, FESCA 2014, Grenoble, France, 12 April 2014, pp. 32–46 (2014)
7. Larsen, P.G., Battle, N., Ferreira, M., Fitzgerald, J., Lausdahl, K., Verhoef, M.: The overture initiative integrating tools for vdm. SIGSOFT Softw. Eng. Notes **35**(1), 1–6 (2010)
8. Liskov, B.H.: A design methodology for reliable software systems. In: Proceedings of the 5–7 December 1972, Fall Joint Computer Conference, Part I, AFIPS 1972 (Fall, part I), pp. 191–199. ACM, New York (1972)
9. Martin, V.A., Lycett, M., Macredie, R: Exploring the gap between business and it: an information culture approach. In: Proceeding in Action in Language, Organisations and Information Systems, ALOIS (2003)
10. Myburgh, A.: Situational software engineering complex adaptive responses of software development teams. In: 2014 Federated Conference on Computer Science and Information Systems (FedCSIS), pp. 841–850, September 2014
11. OMG-MDA. Model driven architecture (mda), mda guide rev. 2.0, June 2014. http://www.omg.org/cgi-bin/doc?ormsc/14-06-01.pdf. OMG Document ormsc/2014-06-01
12. The OpenGroup. The IT4IT Reference Architecture, Version 2.0. Number C115 in Open Group Standard. Van Haren, October 2015
13. Osório, A., Camarinha-Matos, L.M., Afsarmanesh, H.: Cooperation enabled systems for collaborative networks. In: Camarinha-Matos, L.M., et al. (eds.) PRO-VE 2011. IFIP AICT, vol. 362, pp. 400–409. Springer, Heidelberg (2011)
14. Osório, L.A., Camarinha-Matos, L.M., Afsarmanesh, H.: ECoNet platform for collaborative logistics and transport. In: Camarinha-Matos, L.M., et al. (eds.) PRO-VE 2015. IFIP AICT, vol. 463, pp. 265–276. Springer, Heidelberg (2015). doi:10.1007/978-3-319-24141-8_24
15. Rendon, R.G.: Using a modular open systems approach in defense acquisitions: implications for the contracting process. In: 2007 IEEE International Conference on System of Systems Engineering, SoSE 2007, pp. 1–6, April 2007
16. Ryan, S., OConnor, R.V.: Acquiring and sharing tacit knowledge in software development teams: an empirical study. Inf. Softw. Technol. **55**(9), 1614–1624 (2013)
17. Soffer, A.: Bridging information gaps in an OPM-based system development lifecycle. In: Meersman, R., Tari, Z., Corsaro, A. (eds.) OTM-WS 2004. LNCS, vol. 3292, pp. 787–797. Springer, Heidelberg (2004)
18. Zahariadis, T., Papadakis, A., Alvarez, F., Gonzalez, J., Lopez, F., Facca, F., Al-Hazmi, Y.: Fiware lab: managing resources and services in a cloud federation supporting future internet applications. In: Proceedings of the 2014 IEEE/ACM 7th International Conference on Utility and Cloud Computing, UCC 2014, pp. 792–799. IEEE Computer Society, Washington (2014)