Architecture for the Use of Synergies between Knowledge Engineering and Requirements Engineering - Extended Abstract *

José del Sagrado, Isabel M. del Águila, and Francisco J. Orellana

Dpt. Languages and Computation,
Ctra Sacramento s/n, 04120 University of Almería, Spain
{jsagrado,imaguila,fjorella}@ual.es

Expert knowledge is involved in every software development project since developers must face numerous decision tasks during requirements management, analysis, design, and implementation stages. Therefore, if expert knowledge could be properly modelled and incorporated in the different processes of software development as well as in the CASE tools that support these processes, that would mean a great advantage for any software development.

In software development, requirements stage is considered a good application domain for Artificial Intelligence (AI) techniques because of requirements nature. Software requirements express and establish the needs and constraints that contribute to the solution of a real world problem [7]. However, requirements tend to be imprecise, incomplete and ambiguous[3] and has a big impact in whole development stages [5, 15, 1, 2]. Therefore, the use of AI techniques in order to improve requirements stage will favorably affect the whole software life cycle, but we need a seamless integration of Requirement Engineering (RE) and AI techniques to exploit the benefits of collaboration between these two knowledge areas [10].

Besides, the biggest breakthrough in requirement management is when you stop thinking of documents and start thinking about information. Here, is where CARE (Computer-Aided Engineering Requirement) tools help us in order to be able to handle all of this information. InSCo Requisite is an academic web CARE tool, developed by DKSE group at the University of Almería, which aids during the requirement development stage [11].

This work presents the architecture for the seamless integration of a CARE tool to manage requirements (i.e. InSCo Requisite) with some AI techniques (i.e. Bayesian networks [12, 6] and metaheuristics). Specifically, a Bayesian network, called Requisites [13], is used in the requirement validation task in order to validate the Software Requirements Specification (SRS) of a software development project, they has been successfully applied in SE, [9, 4, 8, 13]. Metaheuristic techniques (Simulated Annealing, Genetic Algorithms and Ant Colony Systems) are used in the problem of selecting the subset of requirements among a whole

* This is a long abstract of the paper published in Lozano,J.A., Gámez, J.A., Moreno, J.A. (eds.) LNAI Series, Current Topics in Artificial Intelligence. 14th Conference of the Spanish Association for Artificial Intelligence, CAEPIA 2011, La Laguna, Spain, November 8-11, 2011, Selected Papers.

set of candidate requirements proposed by a group of stakeholders, that will be included in the development of a final software product [14].

The RE workflow depicted in Figure 1 shows an organization of the tasks that must be done in a software development project during RE stage. Requirements are elicited or gathered from users, next they are specified in a document or its electronic equivalent, known as Software Requirements Specification (SRS). CARE tools provide environments that make use of databases, allowing an effective management of the requirements of any software project. Requirements validation checks whether the elicited and specified requirements present inconsistencies; if the information is incomplete or if there are ambiguities in the system definition. Requisite Bayesian network provide developers an aid, under the form of a probabilistic advice (i.e. an estimation of the degree of revision for the SRS), helping them at the time of making a decision about the stability of the current requirements specification. Finally, requirements selection task has as main objective to choose, from all the requirements defined in the specification, the subset of requirements that will be implemented.

Bayesian networks and metaheuristic techniques have demonstrated to obtain interesting results through different tests data [13, 14]. However, it is difficult to put them in practice in real software projects. We strongly believe that having these AI techniques available in a CARE tool would be considerably helpful for any development team, making them more accessible even for non-expert people. However, IA techniques and the CARE tools have been developed independently of each other. Therefore, it is necessary to define a communication interface between them preserving the independent evolution of both areas and achieving a synergic benefic effect between them. This seamless synergic architecture is shown in Figure 1. The architectural pattern distinguish between three logically separated layers (see Fig. 1) : the presentation (i.e. interface layer), the application processing (i.e. service layer), and the data management (i.e. data layer).

The interface is a web environment accessed from a web browser. Data layer is in charge of storing and managing the electronic representation of SRS handled by InSCo Requisite tool and the knowledge base that contains the Bayesian network Requisites. Service layer is composed by the CARE tool (i.e. InSCo Requisite), the AI techniques used to address requirements validation (i.e. Bayesian network Requisites) and requirements selection (i.e. metaheuristics algorithms) tasks. Communication interface connect CARE and knowledge-based tools passing the required information needed for the execution of the appropriated processes. Thus, requirement validation receives metrics on the SRS and returns an estimation of the degree of revision for SRS; requirement selection receives resources effort bound and specific measures on individual requirements and set the set of requirements in order to be implemented. All of these communication processes are performed through XML files.

The purpose of this work is to define a three-layer architecture which: a) allows the seamless collaboration between RE tasks and some AI techniques (Bayesian networks, simulated annealing, genetic algorithms and ant colony sys-

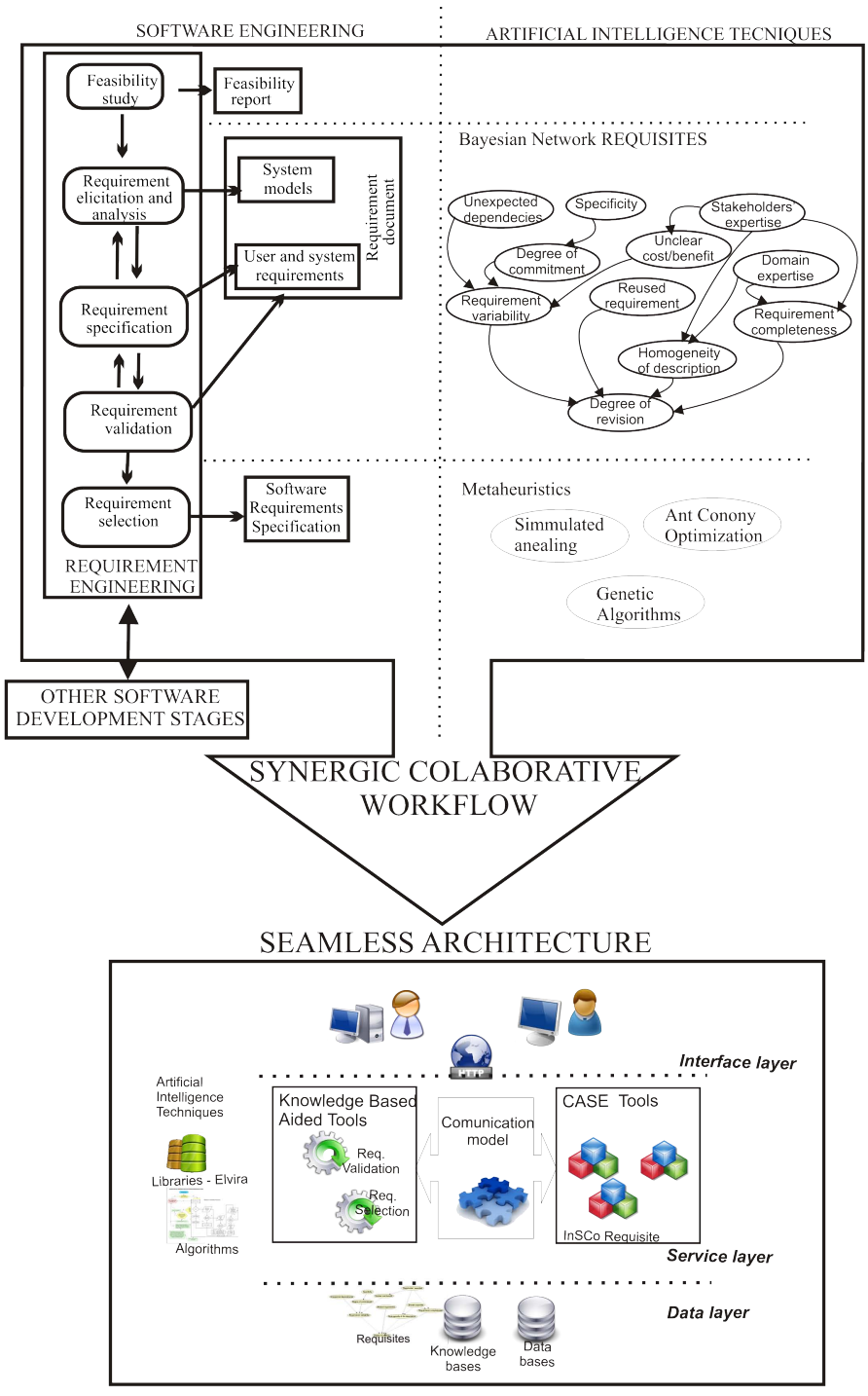

Fig. 1. Seamless synergic architecture.

tems) in order to perform a software development project; b) facilitates their parallel and independent evolution.

Acknowledgments. This work was supported by the Spanish Ministry of Science and Innovation under project TIN2010-20900-C04-02 and by the Junta of Andalucía under project TEP-06174.

References

1. Standish Group: Chaos Report. Technical report, Standish Group International (1994)
2. Johnson, J.: CHAOS chronicles v3.0. Technical report, Standish Group International (2003)
3. Cheng, B.H., Atlee, J.M.: Research directions in requirements engineering. In: Future of Software Engineering, FOSE'07, pp. 285-303. Institute of Electrical and Electronics Engineers, Minneapolis, Minnesota (2007)
4. Fenton, N., Neil, M., Marsh, W., Hearty, P., Marquez, D., Krause P., Mishra, R.: Predicting software defects in varying development lifecycles using Bayesian nets, Information and Software Technology 49(1) 32-43 (2007)
5. Glass A.R.L.: Facts and Fallacies of Software Engineering. Pearson Education, Inc.,Boston, MA (2002)
6. Jensen F.V.: Bayesian Networks and decision graphs. Springer-Verlag, New York (2001)
7. Kotonya, G., Sommerville, I.: Requirements Engineering: Processes and Techniques. Wiley (1998)
8. Lauria, E.J.,Duchessi, P.J., A Bayesian Belief Network for IT implementation decision support. Decision Support Systems 42(3), 1573-1588 (2006)
9. de Melo A.C. , Sanchez, A.J.: Software maintenance project delays prediction using Bayesian Networks. Expert Systems with Applications 34(2), 908-919 (2008)
10. Meziane, F., Vadera, S. (eds.): Artificial intelligence applications for improved software engineering development: new prospects. IGI Global, Hershey, New York (2010)
11. Orellana, F.J., Cañadas, J., del Águila, I.M., Túnez, S.: INSCO requisite - a Web-Based RM-Tool to support hybrid software development. In: International Conference of Enterprise Information System ICEIS (3-1), pp. 326-329. Barcelona, Spain (2008)
12. Pearl, J.: Probabilistic reasoning in intelligent systems: networks of plausible inference. Morgan Kaufman, San Mateo, CA (1988)
13. del Sagrado, J. , del Águila, I.M.: A Bayesian Network for Predicting the Need for a Requirements Review. In: Meziane, F., Vadera, S. (eds.): Artificial intelligence applications for improved software engineering development: new prospects, pp. 106-128. IGI Global, Hershey, New York (2010)
14. del Sagrado, J., del Águila, I. M., Orellana, F. J.: Requirement selection: Knowledge based optimization techniques for solving the next release problem. In: 6th Workshop on Knowledge Engineering and Software Engineering (KESE 2010), pp. 40-51. CEUR-WS, Karlsruhe, German (2010)
15. Sommerville I.: Software Engineering. Addison-Wesley Longman Publishing Co., Inc., Boston (2006)