Modeling Tool for Managing Requirements and Backlogs in Agile Software Development

Cătălina Floruț [1] and Robert Andrei Buchmann [1]

[1] *University Babeș-Bolyai, Faculty of Economics and Business Administration, Business Informatics Research Center, str. T. Mihali 58-60, Cluj Napoca, 400591, Romania*

Abstract

Context and motivation. Jira is an established tool for issue tracking, requirements and backlog management. Its popularity has impact on how agile software projects are conceptualized by project team members, often dictated by the data structures being managed within the tool. *Question/problem.* Although user stories should be traceable to both software artifacts and business processes, both sides being well served by modeling standards (BPMN, UML), the diagrammatic semantics are not leveraged by issue tracking tools; conversely, modeling tools have been more interested in code generation than in supporting issue trackers and agile practices.
Principal ideas/results. We propose a diagrammatic alternative to Jira, a modeling tool built around a domain-specific modeling language that integrates Agile Software Project concepts with modeling standards (BPMN, UML), while also ensuring interoperability with Jira. The engineering effort was based on the metamodeling approach known as Agile Modeling Method Engineering.
Contribution. The DSML underlying the tool is a conceptual bridge between modeling standards and Jira, and it is developed as a Design Science treatment, iteratively increasing its technological readiness.

Keywords

Agile project management, Metamodeling, Domain-specific Conceptual Modeling, ADOxx, User story conceptualization

1. Introduction

Theory and practice of agile software development are sometimes diverging – while the agile paradigm's strongly principled foundations introduced its key concepts (e.g. epic, story), the actual operationalization of those concepts is often distorted by the employed tooling. For example, the concept of "epic" was introduced in Cohn's seminal book [1] as a user story too large to be easily managed, i. e. determined by sizing and complexity and requiring further slicing; however, issue trackers treat it as a level of aggregation, a way of grouping user stories for reporting purposes - a potential obstacle in the way of vertical story slicing [2]. The original conceptualization is flattened or oversimplified by tooling decisions – e.g., in different contexts an epic may be considered a type of backlog item, a large user story, a "container of initiatives" [3]. Although the user story's structure is well known ("as a ...., I want ... because ..."), in practice they are sometimes conflated with "features" or even "tasks" – an oversimplification witnessed first-hand by the authors of this paper in the local industry, which triggered the development of the tool hereby reported.

Domain-specific metamodeling tailors granular and rich conceptualizations for specialized purposes and fields of activity, providing design spaces that reflect those conceptualizations. This paper promotes

In: J. Fischbach, N. Condori-Fernández, J. Doerr, M. Ruiz, J.-P. Steghöfer, L. Pasquale, A. Zisman, R. Guizzardi, J. Horkoff, A. Perini, A. Susi, M. Daneva, A. Herrmann, K. Schneider, P. Mennig, F. Dalpiaz, D. Dell'Anna, S. Kopczyńska, L. Montgomery, A. G. Darby, and P. Sawyer (eds.): Joint Proceedings of REFSQ-2022 Workshops, Doctoral Symposium, and Poster \& Tools Track, Birmingham, UK, 21-03-2022, published at http://ceur-ws.org
EMAIL: catalinaiulia.florut@gmail.com (C. Florut); robert.buchmann@ubbcluj.ro (R. A. Buchmann)
ORCID: 0000-0002-7385-1610 (R. A. Buchmann)

a diagrammatic modeling alternative to Jira that captures a superseding conceptualization of Agile Software Projects assets (particularly epics, stories, sprints, backlog items) that also incorporates attributes proposed by theoretical frameworks. On the conceptualization side it ensures the contextualization of those assets with established modeling languages (BPMN, UML); on the operationalization side, it provides basic interoperability with Jira.

2. Background, Motivation and Novelty

Domain-specific modeling languages can play a key role in requirements traceability [4]. Modeling methods have been proposed for various aspects of requirements engineering, however not in direct relation to first-class concepts of agile practices – e.g. the i* language [5], the User Requirements Notation (URN) [6]. Closer to our approach (also in the choice of metamodeling technology) is the User Story Mapping method [7] – however it does not report interoperability features, as it focuses on ontology-based annotations. Repurposing of conceptual modeling for specific forms of requirements gathering (e.g. mind maps) have been reported in [8], without being interested in interoperability with issue trackers. Requirements interdependencies have been recognized for a long time but little was done in terms of a metamodeling treatment of those dependencies and how they could be traced to business processes. The Attlasian Marketplace offers extensions for Jira diagrammatic documentation [9] without the granular metamodel-level integration proposed in this work.

Our approach is to turn to AMME - Agile Modeling Method Engineering (methodology detailed in the context of a previous project in [10]) in order to build a domain-specific modeling tool that creates a layer of abstraction over Jira and over the Agile Software Development domain. In terms of discourse domain, we're primarily referring to [1] and subsequent frameworks such as [3], since earlier attempts (even the Agile Manifesto) are abstract and principle-oriented rather than outlining first-class conceptual citizens. Compared to what established modeling standards offer, domain-specific modeling aims for conceptualizations of narrower scope and deeper specialization, aiming to support a domain of activity or a narrow technological specificity. For the end-user, the tool provides a visual management approach of assets that are traditionally handled with Jira - here also contextualized with business process management and systems architecture aspects; therefore, the proposal is bridged with BPMN and UML with the help of a metamodeling platform (ADOxx [11]).

3. Use Cases and Implementation Details

The engineering method (right side of Figure 1) started from an available open source BPMN+UML implementation in the BEE-UP tool [12, 13], made available by the OMiLAB digital innovation ecosystem [14]. ADOxx [11] was used to extend the conceptualization of BEE-UP to obtain a modeling tool supporting the use cases depicted in Figure 1 (left side).

The user experience and look/feel are inherited from ADOxx and it's typical for a diagrammatic modeling tool (drag and drop, as well as editing of machine-readable property sheets and hyperlinks for each individual model element). The internal scripting language of ADOxx was employed to emulate code generation, repurposed to generate data structures that are importable by Jira. Since ADOxx manipulates a graph-like structure of diagrammatic models interconnected by semantic links, the new concepts enable certain traceability features (model queries) for the new conceptualization. The semantic links are set by the modeler in the same property sheet like other attributes, with the additional possibility of targeting model elements of entire models, of types constrained by the metamodel design.

Taking cues from multi-perspective enterprise modeling [15], the proposed modeling language is partitioned in two viewpoints – structural and behavioral. The *structural part* (Figure 2) was developed initially to mirror data structures handled inside Jira (i.e. project decomposition), later enriched with concepts from theoretical frameworks (SAFe [3]) and finally extended with semantic links to UML concepts available in the BEE-UP tool. UML already provides diagrams to decompose a system architecture (deployment diagram, component diagram) or to capture stakeholder expectations (use cases as proxy for goals). The tool ensures that the rich conceptualization of UML becomes a semantic context for backlog items typically managed through Jira – e.g. the three segments of a user story (*As a ... I want to ... in order to ...*) become a machine-readable structure acting as a traceable n-ary

relationship between stakeholders, expectations (from use case diagrams), business processes (BPMN) and architectural elements (UML).

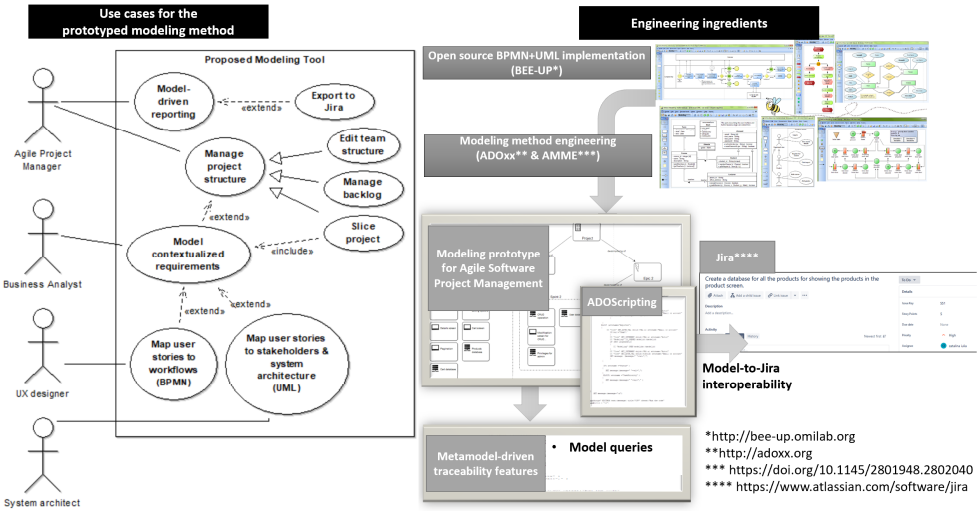

Figure 1: Use cases for the proposed tool (left) and engineering components (right).

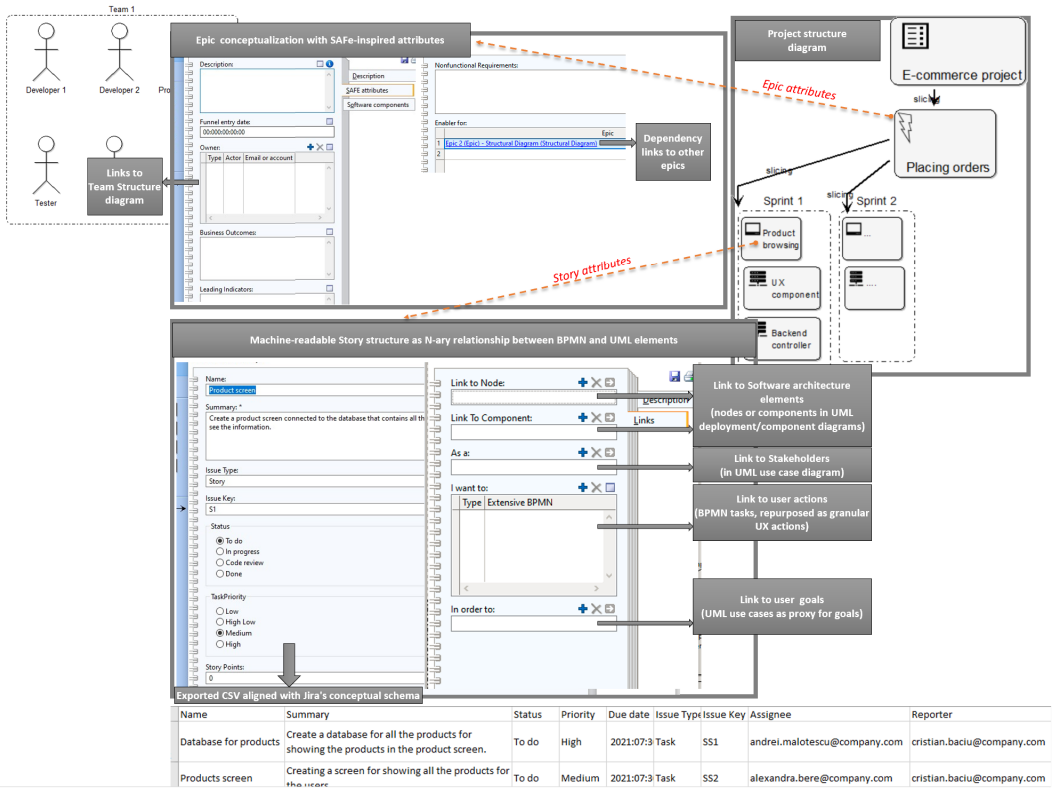

Figure 2: Structural view on an agile software project with semantic links and Jira interoperability

The *behavioral part* (Figure 3) covers two perspectives: (a) *Customer-oriented:* business processes or user experience granularly mapped on user stories and other backlog items. User stories are part of user experience processes which in turn are contextualized by higher level business processes; Business Analysis or Requirements Engineers with a process thinking mindset typically collect or design business process models to provide context or explainability for requirements. The hereby proposed tool makes this traceable on a machine-readable level, beyond the story dependencies typically handled by issue trackers. The semantic links can exist in both directions between BPMN and user stories – a

complex task could be decomposed in granular stories detailing who should do what; or, a user story could be detailed in a BPMN diagram fully depicting the user experience (e.g. sequence of clicks).

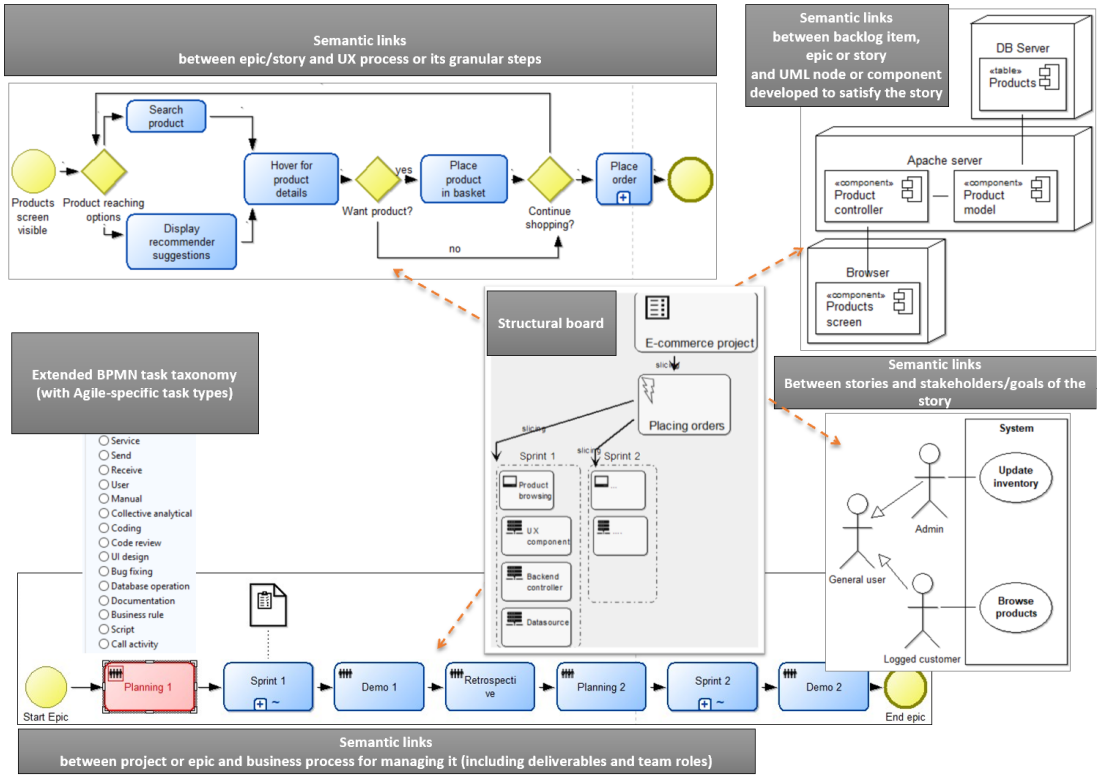

Figure 3: Behavioral perspectives: Development process for an epic (bottom) and UX process (top). Also shown, an extended BPMN task taxonomy and links to UML (stakeholders, software components)

A Project-oriented behavioral perspective gives to the agile project manager the possibility to use BPMN (with a custom task taxonomy) for describing processes that are internal to the software development company, beyond the barebone Jira issue lifecycle; this allows to leverage the possible BPM background of a project manager (possibly developed during a Business Analysis phase) – software companies are not absolved of monitoring and managing their own internal processes.

The export to Jira is realized in the internal ADOxx scripting language by reading diagrammatic contents from the currently opened model and across its semantic links, to generate a CSV file that can be imported by Jira. The scripting flavor is suggested in the code samples listed in Figure 4.

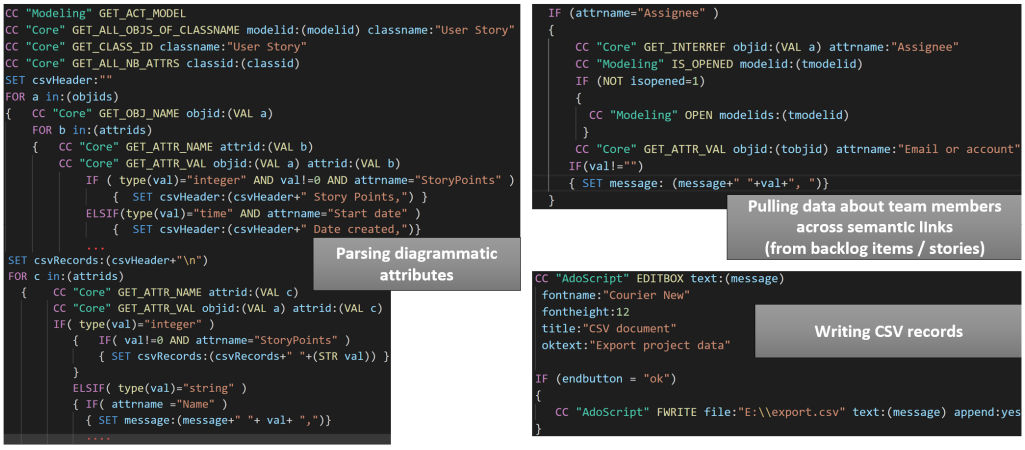

Figure 4: ADOscripting for parsing diagrammatic content to the CSV structure expected by Jira

4. Evaluative Plan

Considering the taxonomy of Design Science artifact evaluation criteria [16], we target the following:

A. Evolution ("learning capability") is the reason for adopting the AMME methodology to gradually capture richer conceptualizations as feedback from practitioners. The tool's agile evolution owes to the iterative nature of the AMME methodology and the fast-prototyping capabilities of the ADOxx platform. The hereby presented artifact is already in its third evolutionary iteration:

a. the original one was limited to making possible the export from a diagrammatic structure to Jira's expected data structure;
b. the second iteration evolved the conceptualization beyond Jira's required input;
c. the current iteration added the BPMN and UML integration benefitting from the already existing open BPMN/UML implementation available in the BEE-UP tool [12,13];
d. the backlog for future iterations includes (i) streamlined interoperability with Jira via HTTP (instead of the intermediate file format); (ii) conversion of a project structure to a Kanban board allowing users to switch between Scrum and Kanban practices.

B. Consistency with technology ("harnessing existing technology") – one key goal was to ensure interoperability with the most popular issue tracker used at the authors' workplace. Although the diagrammatic tool provides a richer conceptualization, it supersedes and extends Jira concepts, rather than disrupting familiarity. The extent to which this familiarity is disrupted will require user-based evaluation approaches that have been established for AMME-driven tools [17]:

a. evaluation of modeling task performance: observing how subjects already familiar with setting up a Jira project describe the same project in a diagrammatic way, after a brief training;
b. interviews to understand the obstacles and mistakes in performing the modeling task – these may range from lack of semantic transparency (for the choice of graphical symbols or concept labelling), to the absence of concepts and properties forcing users to textually annotate models when they cannot express certain details.

Selection of subjects for such protocols must take into consideration experience with Jira and Scrum for most modeling tasks, and with BPMN/UML modeling for when the BPMN/UML contextualization is applied to project components and backlog items.

C. Completeness relative to a small set of traceability requirements, satisfied through the metamodel-driven model query engine [18] allowing the navigation of semantic links and retrieval of backlog items' contextual information, e.g.

... retrieving the epics to which a particular controller component (backlog item) belongs: *(((<"Products controller"><--)<-"Is inside")<-"slicing")>"Epic"<*

... retrieving the stakeholders, whose user stories are satisfied by a particular software component *((<"Products controller"><--)-->"As a")>"Stakeholder role"<*

Such requirements evolve and their evolution is directly addressed by the first criterion discussed here: if these queries are seen as a proxy for "competency questions" (as understood in the field of ontology engineering), then the ability to satisfy new queries is achieved by enriching the conceptualization on metamodeling level, in order to accommodate the level of specificity needed for these queries. As a concrete example, the second query shown above was only possible in the 3rd iteration of the tool, when semantic linking to UML actors was enabled for tool users.

5. Limitations and Outlook

An intrinsic limitation is that most modeling will be perceived as overhead effort for agile project managers or for those inexperienced with diagrammatic modeling. A learning curve supported by a documented modeling procedure is necessary to bridge the Scrum mindset with the BPMN/UML modeling mindset - however there's no reason why these should be disjoint disciplines since business process knowledge is relevant on both sides of a software project - customer side (as user experience context) and project management side (as business processes for the software development business).

The current technical limitations are also backlog items for the future development of the tool: (a) interoperability with Jira is currently achieved only through the CSV import-export mechanism, but Jira also supports HTTP connectivity, which could streamline the communication between the modeling

tool and a Jira server; (b) alternatives to Jira are also considered - experience with FusionForge is also available to authors; (c) the tool is currently Scrum-centered, but an automatic generation of a Kanban visual board is also feasible.

6. References

[1] M. Cohn, User Stories Applied: For Agile Software Development, Addison-Wesley, 2004.

[2] M. Riley, User stories: A tale of epic confusion, 2020, URL: https://www.thoughtworks.com/insights/blog/user-stories-tale-epic-confusion

[3] Scaled Agile, Epic-Scaled Agile Framework, 2021. URL: https://www.scaledagileframework.com/epic/

[4] M. Taromirad, R. F. Paige, Agile requirements traceability using domain-specific modelling languages, in: Extreme Modeling Workshop, XM 2012 - Satellite Event of the IEEE/ACM 15th International Conference on Model Driven Engineering Languages and Systems, 2012, pp. 45-50, doi: 10.1145/2467307.2467316

[5] E. Yu, J. Mylopoulos, Understanding 'why' in software process modelling, analysis, and design, Proceedings of ICSE 1994, IEEE, 1994, pp. 159-168, doi: 10.1109/ICSE.1994.296775

[6] D. Amyot, Introduction to the User Requirements Notation: learning by example, in: Computer Networks, 2003, pp. 285-301, doi: 10.1016/S1389-1286(03)00244-5

[7] D. Kiritsis, A. Milicic, A. Perdikakis, User Story Mapping-Based Method for Domain Semantic Modeling, in: Domain-Specific Conceptual Modeling: Concepts, Methods and Tools, Springer, 2016, pp. 439-454, doi: 10.1007/978-3-319-39417-6_20

[8] R. A. Buchmann, A. M. Ghiran, C. C. Osman, D. Karagiannis, Streamlining Semantics from Requirements to Implementation Through Agile Mind Mapping Methods, in: Proceedings of REFSQ 2018, Springer, 2018, pp. 335-351, doi: 10.1007/978-3-319-77243-1_22

[9] Atlassian Marketplace, Draw.io for Jira, 2022, URL: https://marketplace.atlassian.com/apps/1211413/draw-io-diagrams-for-jira

[10] R. A. Buchmann, D. Karagiannis, Agile Modelling Method Engineering: Lessons Learned in the ComVantage Research Project, in: Proceedings of PoEM 2015, Springer, 2015, pp. 356-373, doi: 10.1007/978-3-319-25897-3_23

[11] BOC GmbH, The ADOxx Metamodelling Platform, 2022, URL: https://www.adoxx.org/live/home

[12] OMILAB.org, Bee-Up for Education, 2022, URL: https://bee-up.omilab.org/activities/bee-up/

[13] D. Karagiannis, R. A. Buchmann, P. Burzynski, U. Reimer, M. Walch, Fundamental Conceptual Modeling Languages in OMiLAB, in: Domain-Specific Conceptual Modeling: Concepts, Methods and Tools, Springer, 2016, pp.3-30, doi: 10.1007/978-3-319-39417-6_1

[14] D. Karagiannis, R. A. Buchmann, X. Boucher, S. Cavalieri, A. Florea, D. Kiritsis, OMiLAB: A Smart Innovation Environment for Digital Engineers, in: Proceedings of PRO-VE 2020, Springer, 2020, pp. 273-282, doi: 10.1007/978-3-030-62412-5_23

[15] U. Frank, Multi-perspective enterprise modeling: foundational concepts, prospects and future research challenges, in: Software & Systems Modeling 13, 2012, pp. 941-962, doi: 10.1007/S10270-012-0273-9

[16] N. Prat, I. Comyn-Wattiau, J. Akoka, Artifact evaluation in information systems design-science research – a holistic view, in Proceedings of PACIS 2014, Association for Information Systems, paper 23, URL: https://aisel.aisnet.org/pacis2014/23/.

[17] Buchmann, R.A., Karagiannis, D.: Modelling mobile app requirements for semantic traceability, Requirements Engineering 22(1):41–75, 2017, doi: 10.1007/s00766-015-0235-1

[18] BOC GmbH, Query Language AQL, 2022, URL: https://www.adoxx.org/live/adoxx-query-language-aql