xtUML: Current and Next State of a Modeling Dialect

(Experience Report and Historical View)

Cortland Starrett
One Fact Inc.
10412 US HWY 52 S
Clarks Hill, Indiana 47930 USA
cortland.starrett@onefact.net

***Abstract*— xtUML and its Papyrus-xtUML (BridgePoint) tooling have advanced further in the last two years than in the ten preceding years.**

Acceleration in usage and improvements in tooling have been fueled by the open source software (OSS) ecosystems of Apache, Eclipse and Papyrus. Development teams have transformed from licensed users into user-contributors.

Executable Translatable Unified Modeling Language (xtUML) is descended from Shlaer-Mellor Object Oriented Analysis (SM-OOA) and has survived the Method Wars, notation unification (UML) and tooling evolution. Sparse but focused deployment in the embedded control and distributed processing worlds has produced hundreds of years of accumulated application model intellectual property (IP).

This paper summarizes a brief history, current status and future of xtUML. It highlights recent advancements, forward direction, key players, related modeling dialects, architectural migration and implications to the metamodels and compliance with Papyrus Platform, UML2, fUML and Alf.

Keywords*— *xtUML, Shlaer-Mellor, Executable UML, UML-RT, MASL

I. Introduction

Executable Translatable Unified Modeling Language (xtUML) [1] is a modeling dialect that employs standard UML notation to express executable models following the Shlaer-Mellor Method of Object Oriented Analysis and Design [3,4]. The method is well-defined and documented and carries a substantial base of research, education and industry usage through the last two decades.

Shlaer-Mellor xtUML has enjoyed evolution and maturity while remaining true to the original method. The history, status and comparison to other dialects are outlined. The future of xtUML is predicted in light of the current players and the positive network effects of Eclipse, Papyrus and the open source community.

It is observed that a fully open source governance and ecosystem around xtUML has dramatically increased the pace of advancements in the tooling and facilitated collaboration among users, suppliers and academics. This is because openness, transparency and elimination of exclusive ownership fosters an environment of security. There is no single entity (licensing company) that must be trusted to serve the interests of all parties. Users can add features (or pay suppliers to add features) at will. In the xtUML community, features and fixes are being delivered faster than ever before. Related executable modeling languages and technologies are sharing in these improvements.

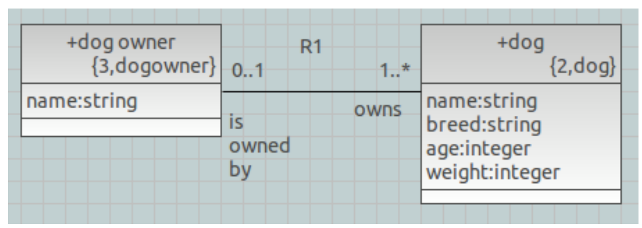

Figure 1: dog owner *owns* dog

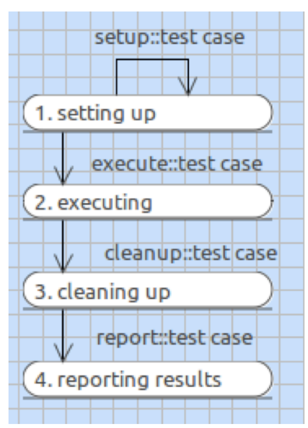

Figure 2: The test case state machine transitions from *setting up* to *executing* to *cleaning up* to *reporting results.*

II. Background

The Shlaer-Mellor xtUML method steps modelers through a sequence of abstract modeling constructs addressing **data**, **control** and **processing** views of systems applications. Data is modeled using class diagrams, control is modeled as state machines and processing is modeled with action language.

The order of modeling encourages as much information as reasonable to be captured in data with the exposure of abstractions at the highest possible level. The method is considered object-oriented due to its emphasis on data modeling. This object concept emphasizes relations between the data abstractions. UML *class diagrams* provide the notational richness required to capture clear abstractions of conceptual entities with classes, attributes and various forms of associations relating them. See Figure 1 for an example class diagram depicting the relationship between a dog and its owner. UML *state machine diagrams* (Figure 2) formalize the lifecycles of individual UML classes. The example illustrates the states of a test case. Thus, concurrent sequential processing is captured in a plurality of relatively simple, communicating instance-based state machines. Finally, activity semantics are modeled in class operations, state machine states and transitions, and function bodies using an abstract *action language* that is Turing Complete but platform independent.

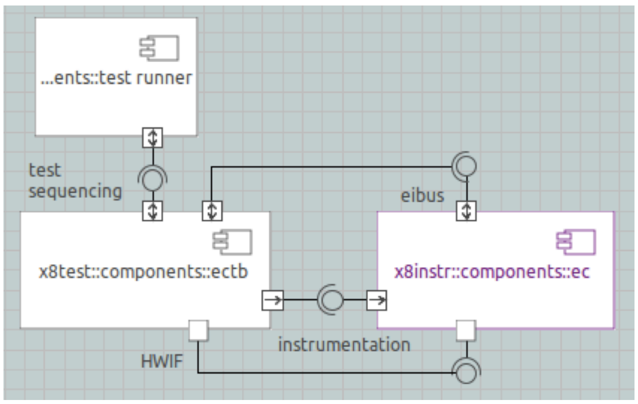

Figure 3: Component Diagram providing functional partitioning

Models are partitioned along subject matter boundaries and deployed as compositions using UML *component diagrams* (Figure 3). The example shows the functional partitioning of an electrochemical (EC) analyzer and an instrumentation test runner which can stimulate and measure its response.

xtUML models are interpretively executable following a set of rules. These rules enforce the semantics of the model artifacts and establish a basis to govern time, order and priority. Execution can be performed in simulations run by humans enforcing the rules or by an xtUML interpreter that automates the execution for model testing purposes.

A corrolary set of semantics governs the tranformation of xtUML from one representation into semantically equivalent forms represented in lower-level target deployment languages such as Ada, SPARK, Java, C, C++, MISRA-C, AUTOSAR, SystemC, VHDL, (IDL, DocBook, HTML). The process of translating xtUML into other forms is called *model compilation* and is performed by a *model compiler*. A model compiler is a refinement of code generation in its complete and strict mapping of the semantic rules of the language. A model compiler must guarantee adherence to semantics to preserve execution behavior between forms. Model compilers can translate only from a higher level of abstraction to a lower level (or the same level). Model compilers can insert additional platform-specific detail into the transformation output. Shlaer-Mellor xtUML model compilers translate PIMs (Platform Independent Models) to PSMs (Platform Specific Models).

III. Brief History

- 1988, 1991 Shlaer-Mellor Method published by Stephen Mellor and Sally Shlaer in [3,4].
- 2002 Executable UML established as Shlaer-Mellor OOA using UML notation [14].
- 2013 BridgePoint xtUML Editor goes open source under Apache 2.0 [1].
- 2014 all of BridgePoint (including Verifier and model compilers) goes open source under Apache and Creative Commons.
- 2015 Papyrus Industry Consortium [6] and xtUML/BridgePoint contribution
- 2015 OSS of alternate generator engine [13]
- 2016 Papyrus-xtUML (BridgePoint) Eclipse Foundation governance
- 2016 OSS contributions from industry, university and individuals

Software is "eating the world" [8]. Open source governance is a requirement for modeling tools in order to enable collaboration between users, suppliers and academia [9].

The establishment of open source software and transparent governance has triggered a massive influx of openly accessible models, tooling source code, documentation, research results, education materials and supporting experience results. These contributions are coming from Europe, Asia and North America. Communities spanning these continents are collaborating at ever increasing rates due to the network effects and accessibility.

IV. Key Players

xtUML is driven by several factors. The tooling would seem to be the obvious driver, but the user base and the collection of intellectual property (IP) are producing the real energy. IP accumulated over decades provides a base. The pool of existing models is continually upgraded and expanded as the language and technology evolve. The cache of existing models is extended as applications gain features and heritage functionality is reused in fresh contexts. Models are beginning to be exchanged more openly in shared repositories encouraging reuse and providing a starting point for new modelers [24]. This base also serves to vet and test new and updated tools.

Following are key user teams that influence the language and tools of xtUML.

A. Saab

Saab [20] has been a user, advocate and conference presenter on the application of xtUML in MDA/MDE contexts since 2000. Primary motivations for adopting the Shlaer-Mellor Method are the interpretive executability including action semantics (action language) and model compilation. The company has applied the technology in aviation, military and three-dimensional graphics mapping systems. Beyond application IP, Saab has leveraged open model compilation to produce Ada, SPARK, C and VHDL from xtUML models. Saab is now an advocate of open source modeling as part of the Papyrus Industry Consortium [6].

B. *Fuji-Xerox*

Fuji-Xerox [21] has about the same duration of experience applying xtUML in office automation equipment (combined copier/printer/scanner). Engineers at Fuji-Xerox have been instrumental in taking executable modeling out of the workplace and into education. Fuji-Xerox has sponsored and assisted teams of students and engineers in contests such as the ET-Robocon [15] where robots powered by xtUML have won national championships. Example models and target development environments have been configured and contributed to public git repositories [25].

C. *Ericsson*

Ericsson [22] has pioneered Executable UML research and tools development. In the realm of model compilers, Ericsson built a C model compiler that generates efficient code for cellular network nodes. Ericsson invests heavily in open source modeling and serves in the Papyrus Industry Consortium. xtUML was attractive because of the focus on data modeling and open, metamodel-based model compilation.

D. *UK Crown*

Agencies within the United Kingdom have been active Shlaer-Mellor modelers for most of two decades. One group recently migrated models from a commercial proprietary UML tool to Papyrus-xtUML (BridgePoint) and contributed a C++ model compiler targeted at distributed POSIX. Additionally, a run-time model debugger was added to the open source repository of xtUML tools.

E. *Agilent*

A group within Agilent [23] formerly known as Varian has developed a line of gas chromatography equipment using xtUML. These devices have been running in the field for more than ten years with new, modeled features added by teams of modelers in the USA, Netherlands and China. The ability to preserve the application PIM and modify only the PSM in the model compiler enabled Agilent to retarget as electronics and microcontrollers advanced.

F. *Research and Academia*

xtUML and Papyrus-xtUML (BridgePoint) have been used in curricula at the university and secondary school levels. Examples include universities in Japan (Shinshu University, Kyushu University), Croatia (University of Zagreb), Hungary (Budapest University of Technology and Economics), Sweden (Chalmers of Technology) and a high school in the United States (Faith Christian) [16]. Courses using xtUML often include hands-on modeling, execution and translation onto targets. Target environments have included various robotics packages and small embedded control boards that provide physical evidence of execution behavior.

V. Current State

A. *Body of Intellectual Property*

The current body of intellectual property encoded as xtUML models is rich and growing. One company reports an investment of four hundred years of engineering in applications employing the methodology. A recent trend is that more models are shared openly [24]. This gives new projects a head start and encourages reuse among all teams.

B. *Self Hosting*

One xtUML model of particular interest that is openly shared is the model of Papyrus-xtUML (BridgePoint) itself. The tool is simply an extension of the xtUML metamodel. Editor, Execution and Translation functionality are xtUML application domains built on top of the "OOA of OOA". The whole is compiled with a Java model compiler and packaged as Eclipse plugins. Papyrus-xtUML (BridgePoint) is a modeling tool that models and compiles itself.

C. *Papyrus and Papyrus Industry Consortium*

xtUML tooling leverages underlying infrastructure based on Java and Eclipse. Work is ongoing to establish greater commonality and standardization by deploying EMF [18] and Papyrus [5]. The Papyrus Platform has gained widespread acceptance as an open standard for UML2-based modeling and is being sponsored and promoted by participants of the Papyrus Industry Consortium (Figure 4).

VI. Related Modeling Dialects

It is informative to compare and contrast xtUML with other UML-based executable modeling languages. Below are a few dialects of particular interest.

A. *MASL*

MASL [26] is a human readable concrete textual syntax for a Shlaer-Mellor derived modeling language. MASL describes both structure (classes and state machines) as well as activities (action language) in one cohesive textual language. MASL can be used to express the full semantics of graphical models and has become part of xtUML.

B. *Alf*

Alf [12] is the Object Management Group standard for expressing activities. Like MASL, Alf can express the structure of an Executable UML model. Alf is interpretively executable and compilable. xtUML is slowly but purposefully converging to the Alf standard. Meanwhile, Alf is evolving to support additional modeling semantics (e.g. state machines).

C. *UML-RT*

UML-RT [17] is based on ROOM [7]. UML-RT is similar to xtUML in its focus on execution and architecture (run-time). The two dialects differ in the underlying methodology. ROOM is predominantly functional decomposition within capsules running hierarchical state machines. UML-RT manages complexity with hierarchical state machines where xtUML focuses on data modeling with relational class diagrams. UML-RT employs a fixed run-time environment where xtUML generates the run-time based on rules defined in the model compiler.

UML-RT is supported on the Papyrus Platform and has paved the way for xtUML to adopt a similar tooling approach built on the UML2 compliant editing platform provided by Papyrus. Papyrus-RT and Papyrus-xtUML (BridgePoint) are being managed in the same project group within the Papyrus Industry Consortium and are sharing underlying tools technology.

VII. Next State

A. Eclipse

Papyrus-xtUML (BridgePoint) has been an Eclipse application since 2004. However, until 2013 the source code for the tooling was proprietary. In 2014 all of the tooling opened up and has been placed under governance of the Eclipse Foundation [2]. Greater involvement in the Eclipse community has increased the visibility and acceptance of xtUML tooling.

Look to see the benefits of Eclipse Foundation governance and a growth of the committer and contributor base during the next months and years.

B. Papyrus

Papyrus provides an open source standardized UML2 editing platform supporting profiles. Several modeling dialects are basing their tooling on the Papyrus Platform. Papyrus supports editors for Information Modeling, SysML, UML-RT and other UML-related idioms. Papyrus-xtUML (BridgePoint) is moving onto the Papyrus Platform to leverage this commonality of underlying technology.

C. Action Language

Action languages are model-aware, abstract, platform independent syntaxes for the specification of processing. Papyrus-xtUML (BridgePoint) supports OAL and recently MASL. Support for Alf is under study. Experiments have demonstrated its feasibility.

The next state for action language is proliferation and pluggability. OAL is established; MASL is joining. Interpretive execution of MASL is expected soon. Support of state machine -enabled Alf will follow.

D. Persistence

By default the Eclipse Modeling Framework (EMF) persists data as metamodel instances encoded with XMI. Parallel efforts in various modeling projects are exploring concrete textual syntaxes for modeling languages including the structural elements. Such an approach serializes the in-memory metamodel into a human readable, understandable, parsable and editable language. This strategy has advantages for editability and configuration management (diff/merge). With the advent of MASL and Alf, Papyrus-xtUML (BridgePoint) is likely to adopt this approach.

E. Hybrid Textual/Graphical Editing

Related to the human readable persistence described above is a hybrid approach to editing both graphics and text. Graphics are being recognized as a *view* into the semantics of the model. Therefore, graphical and textual expressions of the meaning of the models can co-exist. Papyrus-xtUML (BridgePoint) can automatically render graphics from textual models. Expect to see advances in model editing that support concurrent textual/graphical editing.

F. Type System

The xtUML type system expressed in the metamodel is largely the same as the original Shlaer-Mellor. Advances proposed by Darwen and Date [19] will drive improvements in the representation of types in xtUML. This type system will elegantly model *types*, *type references* and operations around *compatibility* and *convertability*.

G. fUML and Alf

Members of the xtUML community have been involved in the creation and refinement of the Foundational Executable UML (fUML) [11] and Action Language for Foundational UML (ALF) [12]. However, Papyrus-xtUML (BridgePoint) is minimally compliant to these standards. During migration onto the Papyrus Platform and while changes are made to the xtUML metamodel, conformance to these standards will be a greater priority even as these standards mature and support a greater level of semantics.

VIII. Conclusion

xtUML has remained fundamentally true to the underlying Shlaer-Mellor method through time and tooling evolution. A body of successful projects, models, teams and companies illustrates the strength of this approach to executable modeling. Recent commitments to open source licensing, governance and ecosystems are accelerating advances in the tooling and collaboration between users, suppliers and academics. Specifically, Papyrus and the Papyrus Industry Consortium are bringing together these interested parties in a manner similar to the convergence of technologies in Eclipse. xtUML and executable modeling are benefiting.

References

[1] xtuml.org - Executable, Translatable UML and BridgePoint.

[2] Papyrus for xtUML (BridgePoint) - projects.eclipse.org/projects/modeling.papyrus-xtuml.

[3] Stephen Mellor, Sally Shlaer - Modeling the World in Data, ISBN 978-0136290230.

[4] Stephen Mellor, Sally Shlaer - Object Lifecycles, ISBN 978-0136299400.

[5] Papyrus - projects.eclipse.org/projects/modeling.mdt.papyrus.

[6] Papyrus Industry Consortium - www.polarsys.org/ic/papyrus.

[7] Bran Selic, Garth Gullekson, Paul T. Ward - Real-time Object Oriented Modeling, ISBN 978-0471599173

[8] Mike Milinkovich - Models 2015 Ottawa, www.slideshare.net/eclipsedayindia/eclipse-day-india-2015-keynote-mike-milinkovich

[9] Francis Bordeleau - Models 2015 Ottawa, ceur-ws.org/Vol-1560/keynote.pdf

[10] Unified Modeling Language (UML) - Object Management Group, www.omg.org/spec/UML

[11] Foundational Executable UML (fUML) - Object Management Group, www.omg.org/spec/FUML

[12] Action Language for Foundational UML (ALF) - Object Management Group, www.omg.org/spec/ALF

[13] PyRSL, github.com/xtuml/pyrsl
[14] Stephen Mellor, Marc Balcer, Executable UML: A Foundation for Model-Driven Architectures, ISBN 0201748045
[15] ET-Robocon, www.etrobo.jp
[16] Starrett, Teaching UML Modeling Before Programming at the High School Level, ieeexplore.ieee.org/xpl/login.jsp?tp=&arnumber=4281135
[17] Papyrus for Real Time (Papyrus-RT) - www.eclipse.org/papyrus-rt
[18] Eclipse Modeling Framework (EMF), eclipse.org/modeling/emf
[19] Hugh Darwen, C.J. Date, www.thethirdmanifesto.com
[20] Saab, saab.com
[21] Fuji-Xerox, www.fujixerox.co.jp
[22] Ericsson, www.ericsson.com
[23] Agilent Technologies, www.agilent.com/home
[24] Open Source xtUML models repository github.com/xtuml/models
[25] Open Source models with Japanese language naming github.com/xtuml/modelsjp
[26] MASL – Model Action Specification Language github.com/xtuml/bridgepoint/tree/master/src/org.xtuml.bp.doc/Reference/MASL