

Report on the Tenth International Workshop on Bidirectional Transformations (BX 2022)

Xiao He¹, Li-yao Xia² and Vadim Zaytsev³

¹University of Science and Technology Beijing, China

²University of Edinburgh, United Kingdom

³University of Twente, The Netherlands

Abstract

Bidirectional transformations (BX) are a mechanism for maintaining the consistency between two or more related and heterogeneous sources of information (e.g., relational databases, software models and code, or any other artefacts following standard or domain-specific formats). The strongest argument in favour of BX is its ability to provide a synchronisation mechanism that is guaranteed to be correct by construction. BX has been attracting a wide range of research areas and communities, with prominent presence at top conferences in several different fields (namely databases, programming languages, software engineering, graph transformation). The fast-growing complexity of software- or data-intensive systems forced industry and academia to use and investigate different development techniques to manage many different aspects of the systems. Researchers are actively investigating the use of bidirectional approaches to tackle a diverse set of challenges with various applications including model-driven software development, visualisation with direct manipulation, big data, databases, domain-specific languages, serialisers, data transformation, integration and exchange. BX 2022 is a dedicated venue for BX in all relevant fields and is part of a workshop series that was created in order to promote cross-disciplinary research and awareness in the area.

1. Organisation

The BX workshop series has been running steadily since its conception in 2012 (rotating among ETAPS, STAF, IDBT/ICDT, (Programming), PLW) until 2020 when it was postponed/cancelled together with the rest of STAF 2020 programme due to the pandemic. It has resumed its course in 2021 in a virtual format, and this 2022 edition ran smoothly in a hybrid format, accommodating physical participants in Nantes in France, as well as remote participants via Zoom and Webex. Many names like Frédéric Jouault or Soichiro Hidaka, familiar to the regular BX community but not belonging to on-site attendees, have lighted up on the screen throughout the day and channelled their questions and comments to presenters.

We have received five submissions, and after collecting 2–3 reviews per paper, we ended up with two accepted submissions. Then, we have solicited invited presentations from authors of recently published BX-related papers at ICSE 2022, JSS 189 (2022) and SoSyM 20:5 (2021). The last session of the day was decided to be spent on a plenary discussion.

BX'22: Tenth International Workshop on Bidirectional Transformations, 5 July 2022, Nantes, France

✉ hexiao@ustb.edu.cn (X. He); xialiyao@seas.upenn.edu (L. Xia); vadim@grammarware.net (V. Zaytsev)

🌐 <https://ustbmde.bitbucket.io/hexiao> (X. He); <https://poisson.chat> (L. Xia); <https://grammarware.net> (V. Zaytsev)

🆔 0000-0003-2673-4400 (L. Xia); 0000-0001-7764-4224 (V. Zaytsev)



© 2022 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

CEUR Workshop Proceedings (CEUR-WS.org)

We would like to thank the programme committee members, dedicating their time and expertise to reviewing the submissions:

- ◇ **Jens Weber**, University of Victoria
- ◇ **Michael Johnson**, Macquarie University
- ◇ **Ravi Chugh**, University of Chicago
- ◇ **Perdita Stevens**, The University of Edinburgh
- ◇ **Fernando Orejas**, Universitat Politècnica de Catalunya
- ◇ **Leen Lambers**, Brandenburgische Technische Universität Cottbus-Senftenberg
- ◇ **Hsiang-Shang Ko**, Institute of Information Science, Academia Sinica, Taiwan
- ◇ **Kazutaka Matsuda**, Tohoku University

Our appreciation also goes to all the remote participants who had to bear with the occasional technical imperfections, and especially to physical participants who were literally there for us, and made BX into a successful event with lively discussions and useful insights.

2. Keynote

Zhenjiang Hu is a chair professor in School of Computer Science of Peking University, and a professor of NII by special appointment. He received his B.S. and M.S. degrees from Shanghai Jiao Tong University in 1988 and 1991, respectively, and Ph.D. degree from University of Tokyo in 1996. He was a lecturer (1997–2000) and an associate professor (2000–2008) at University of Tokyo, a full professor at NII (2008-2019), and a full professor at University of Tokyo (2018-2019), before joining Peking University in 2019. His main research interest is in programming languages and software engineering in general, and functional programming and bidirectional programming in particular. He is Fellow of IEEE, Fellow of JFES (Japan Federation of Engineering Society), Member of Engineering Academy of Japan, and Member of Academy of Europe.

Dejima: A Bidirectional Collaborative Framework for Decentralized Data Management

Data management systems are now moving from “centralised” towards “decentralised”, where data are maintained in different sites with autonomous storage and computation capabilities. There are two fundamental issues with such decentralized systems: local privacy and global consistency. By local privacy, the owner of the data wish to control what information should be exposed and how it should be used or updated by other peers. By global consistency, the systems wish to have a globally consistent and integrated view of all data. In this talk, we shall report the progress of our BISCUITS project [1] that attempts to systematically solve these two issues in decentralized systems. In particular, we present a new bidirectional transformation-based approach to controlling and sharing distributed data based on the view, describe Dejima [2], a new architectures for data integration via bidirectional updatable views, and discuss various applications.

The keynote presentation itself covered the history of bidirectional transformations, tracing back to its origins all the way back to VLDB 1978 [3] and POPL 2005 [4] and the seminal Shonan report of 2008 [5]. A fundamental distinction was made between, on one side, forward-based or get-based BX for trees [4], tables [6], strings [7], XML [8] and graphs [9], and, on the other side,

backward-based or putback-based BX [10] for trees [11], tables [12] and graphs [13]. Workshop participants could hear some advice on how to develop dependable, correct-by-construction, BX, with examples from the presenter's own projects like BiGUL [14] and BIRDS [15], as well as the newest of them: Dejima [2].

3. Papers

Weixin Zhang presented *Decomposition Without Regret*, joint work with Cristina Daivid and Meng Wang, and used the well-known Expression Problem [16] to motivate the need to support both object-oriented and functional decomposition. Their tool called Cook, in memory of late William Cook, who contributed heavily to the understanding of abstraction and decomposition [17]. The authors have also extended their talk proposal into a two-page summary which can be found in this STAF 2022 post-proceedings.

Nils Weidmann presented *Bidirectional Transformations in Practice: An Automotive Perspective on Traceability Maintenance*, joint work with Anthony Anjorin and Katharina Artic. Within the context of the automotive industry, they discuss current and potential applications of BX to maintaining models in tandem between requirement management and architectural modelling tools, ensuring their consistency throughout the evolution of a project. Key constraints are identified, stemming from the high-level of abstraction of the models under study and the regulatory nature of the industry. Three main solution strategies are investigated, as a way to classify existing tools and to highlight possible avenues for BX applications. Their nine page submission can also be found in this STAF 2022 post-proceedings.

4. Invited talks

Bernhard Westfechtel presented *BXtendDSL: A layered framework for bidirectional model transformations combining a declarative and an imperative language*, joint work with Thomas Buchmann and Matthias Bank. A common approach in BX is to design domain-specific languages with a strong consistency guarantee between the two directions of a program, but such languages generally sacrifice expressiveness, reducing their practicality. BXtendDSL follows a more pragmatic approach. From a high-level declarative DSL, code is generated against a low-level API, viewed as an imperative DSL. The generated code offers extension points, enabling users to handle operational details that go beyond the scope of the high-level DSL. The resulting language is not only expressive, but also concise and scalable. The six page submission titled *Engineering Bidirectional Model Transformations* that the authors submitted for the STAF 2022 post-proceedings, can be seen as a follow-up to their JSS paper [18]

Gábor Bergmann presented *Controllable and decomposable multidirectional synchronizations*. Whereas BX has traditionally focused on transformations between a source and a single view, this work studies systems with many concurrently synchronised views, called multidirectional synchronisations (MX). This work identifies fundamental challenges in this setting. Notably, a "whack-a-mole" behaviour may arise, where even with individually well-behaved views, local updates can never reach a globally consistent state. Generalisations of standard BX properties as well as novel *axioms of regularity* are investigated. Remarkably, history-ignorance and very-

well-behaved-ness, which coincide for BX, generalise differently for MX. The author decided that his original SoSyM paper [19] does not require a follow-up yet.

Xing Zhang presented *Towards Bidirectional Live Programming for Incomplete Programs*, joint work with Zhenjiang Hu. Bidirectional live programming allows programmers to edit a program by modifying its output. This work extends that idea to also allow editing of incomplete programs. They study a core language for bidirectional live programming featuring explicit holes. During execution, holes in the program become holes in the output. A key mechanism is that output holes also carry the environment in which they were created, allowing one to distinguish different instances of the same hole in the source program. This language is implemented in a tool called Bidirectional Preview, providing a support to perform benchmarks and to demonstrate compelling examples. The authors decided that their original ICSE paper [20] does not require a follow-up yet.

5. Discussion panel

We would like to thank Nils Weidmann, Robbert Jongeling, Massimo Tisi, Soichiro Hidaka, James William Pontes Miranda and Simon Dierl for their contributions to this discussion. All three workshop chairs also participated.

Some of the major discussion points were:

- ◇ The Bx Examples Repository [21] at <http://bx-community.wikidot.com/examples:home>
- ◇ Forward-facing and backward-facing BX
- ◇ Challenges and competitions organised by other communities [22, 23, 24]
- ◇ Possible competition/comparison of approaches or even DSLs
- ◇ Impact of shared cases on replicability if the core of the problem for each case is known and stated
- ◇ Using JANI [25] <https://jani-spec.org> as an interchange format helped the community of probabilistic model checking to foster tool interoperation and comparison, could we at some point converge to such a format for BX?
- ◇ “Editing generated code” as one of the most common BX cases in practice
- ◇ Many alternative possible explorable models of synchronisation with negotiated updates
- ◇ Smell detection in synchronisation problems
- ◇ More demanding properties and stronger guarantees vs realistic properties and weaker guarantees
- ◇ BX in an advisory role to suggest manual changes
- ◇ Maintenance of BX in an industrial context
- ◇ Restoring consistency with machine learning
- ◇ (Partial) bidirectionalisation and defining a bidirectional semantics for a language
- ◇ Impossibility to prove nice theoretical properties in a practical context
- ◇ Using known theory to study non-well-behaved systems
- ◇ Tolerance to imperfect/incomplete practical solutions

References

- [1] Z. Hu, et al., BISCUITS: Bidirectional Information Systems for Collaborative, Updatable, Interoperable, and Trusted Sharing, <http://www.biscuits.work>, 2017.
- [2] K. Miyake, Dejima Prototype, <https://github.com/ekayim/dejima-prototype>, 2020.
- [3] U. Dayal, P. A. Bernstein, On the Updatability of Relational Views, in: Proceedings of the Fourth International Conference on Very Large Data Bases (VLDB), IEEE Computer Society, 1978, pp. 368–377.
- [4] J. N. Foster, M. B. Greenwald, J. T. Moore, B. C. Pierce, A. Schmitt, Combinators for Bi-directional Tree Transformations: A Linguistic Approach to the View Update Problem, in: Proceedings of the 32nd ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages (POPL), ACM, 2005, pp. 233–246. doi:10.1145/1040305.1040325.
- [5] K. Czarnecki, J. N. Foster, Z. Hu, R. Lämmel, A. Schürr, J. F. Terwilliger, Bidirectional Transformations: A Cross-Discipline Perspective, in: Proceedings of the Second International Conference on Theory and Practice of Model Transformations (ICMT), volume 5563 of LNCS, Springer, 2009, pp. 260–283. doi:10.1007/978-3-642-02408-5_19.
- [6] J. N. Foster, T. J. Green, V. Tannen, Annotated XML: Queries and Provenance, in: Proceedings of the 27th Symposium on Principles of Database Systems (PODS), ACM, 2008, pp. 271–280. doi:10.1145/1376916.1376954.
- [7] A. Bohannon, J. N. Foster, B. C. Pierce, A. Pilkiewicz, A. Schmitt, Boomerang: Resourceful Lenses for String Data, in: Proceedings of the 35th Symposium on Principles of Programming Languages (POPL), ACM, 2008, pp. 407–419. doi:10.1145/1328438.1328487.
- [8] D. Liu, Z. Hu, M. Takeichi, An Environment for Maintaining Computation Dependency in XML Documents, in: Proceedings of the Fifth Symposium on Document Engineering (DocEng), ACM, 2005, pp. 42–51. doi:10.1145/1096601.1096616.
- [9] S. Hidaka, Z. Hu, K. Inaba, H. Kato, K. Matsuda, K. Nakano, Bidirectionalizing Graph Transformations, in: Proceeding of the 15th International Conference on Functional Programming (ICFP), ACM, 2010, pp. 205–216. doi:10.1145/1863543.1863573.
- [10] Z. Hu, H. Pacheco, S. Fischer, Validity Checking of Putback Transformations in Bidirectional Programming, in: Proceedings of the 19th International Symposium on Formal Methods (FM), volume 8442 of LNCS, Springer, 2014, pp. 1–15. doi:10.1007/978-3-319-06410-9_1.
- [11] H. Ko, Z. Hu, An axiomatic basis for bidirectional programming, Proceedings of the ACM on Programming Languages 2 (2018) 41:1–41:29. doi:10.1145/3158129.
- [12] V. Tran, H. Kato, Z. Hu, Programmable View Update Strategies on Relations, Proceedings of the VLDB Endowment 13 (2020) 726–739. URL: <http://www.vldb.org/pvldb/vol13/p726-tran.pdf>. doi:10.14778/3377369.3377380.
- [13] X. He, Z. Hu, Putback-based bidirectional model transformations, in: Proceedings of the Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering (ESEC/FSE), ACM, 2018, pp. 434–444. doi:10.1145/3236024.3236070.
- [14] H. Ko, T. Zan, Z. Hu, BiGUL: A Formally Verified Core Language for Putback-based Bidirectional Programming, in: Proceedings of the 21st Workshop on Partial Evaluation and Program Manipulation (PEPM), ACM, 2016, pp. 61–72. doi:10.1145/2847538.2847544.

- [15] V. Tran, H. Kato, Z. Hu, BIRDS: Programming view update strategies in Datalog, Proceedings of the VLDB Endowment 13 (2020) 2897–2900. URL: <http://www.vldb.org/pvldb/vol13/p2897-tran.pdf>. doi:10.14778/3415478.3415503.
- [16] P. Wadler, The expression problem, Posted on the Java Genericity mailing list, <http://homepages.inf.ed.ac.uk/wadler/papers/expression/expression.txt>, 1998.
- [17] W. R. Cook, On understanding data abstraction, revisited, in: Proceedings of the 24th Annual ACM SIGPLAN Conference on Object-Oriented Programming, Systems, Languages, and Applications, OOPSLA 2009, October 25–29, 2009, Orlando, Florida, USA, ACM, 2009, pp. 557–572. doi:10.1145/1640089.1640133.
- [18] T. Buchmann, M. Bank, B. Westfechtel, Bxtenddsl: A layered framework for bidirectional model transformations combining a declarative and an imperative language, Journal of Systems and Software 189 (2022) 111288. doi:10.1016/j.jss.2022.111288.
- [19] G. Bergmann, Controllable and Decomposable Multidirectional Synchronizations, Software and Systems Modeling 20 (2021) 1735–1774. doi:10.1007/s10270-021-00879-w.
- [20] X. Zhang, Z. Hu, Towards bidirectional live programming for incomplete programs, in: Proceedings of the 44th IEEE/ACM 44th International Conference on Software Engineering (ICSE), ACM, 2022, pp. 2154–2164. doi:10.1145/3510003.3510195.
- [21] J. Cheney, J. McKinna, P. Stevens, J. Gibbons, Towards a repository of bx examples, in: Proceedings of the Workshops of the EDBT/ICDT 2014 Joint Conference, volume 1133 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2014, pp. 87–91. URL: <http://ceur-ws.org/Vol-1133/paper-14.pdf>.
- [22] B. E. Aydemir, A. Bohannon, M. Fairbairn, J. N. Foster, B. C. Pierce, P. Sewell, D. Vytiniotis, G. Washburn, S. Weirich, S. Zdancewic, Mechanized metatheory for the masses: The poplmark challenge, in: Proceedings of the 18th International Conference on Theorem Proving in Higher Order Logics (TPHOLs), volume 3603 of *LNCS*, Springer, 2005, pp. 50–65. doi:10.1007/11541868_4.
- [23] D. Beyer, Progress on software verification: Sv-comp 2022, in: D. Fisman, G. Rosu (Eds.), Tools and Algorithms for the Construction and Analysis of Systems, Springer, Cham, 2022, pp. 375–402.
- [24] A. Hartmanns, M. Klauck, D. Parker, T. Quatmann, E. Rujters, The quantitative verification benchmark set, in: Proceedings of the 25th International Conference on Tools and Algorithms for the Construction and Analysis of Systems (TACAS), volume 11427 of *LNCS*, Springer, 2019, pp. 344–350. doi:10.1007/978-3-030-17462-0_20.
- [25] C. E. Budde, C. Dehnert, E. M. Hahn, A. Hartmanns, S. Junges, A. Turrini, JANI: quantitative model and tool interaction, in: Proceedings of the 23rd International Conference on Tools and Algorithms for the Construction and Analysis of Systems (TACAS), volume 10206 of *LNCS*, 2017, pp. 151–168. doi:10.1007/978-3-662-54580-5_9.