John Mylopoulos: Sewing Seeds of Conceptual Modelling

Michael L. Brodie

Verizon Services Operations,
117 West Street,
Waltham, MA 02451-1128, USA
michael.brodie@verizon.com

1 Sewing Seeds of Conceptual Modeling

In the summer of 1980 high in the Colorado Rockies the mountain flowers were blooming, as were ideas of multi-disciplinary conceptual modelling. The Pingree Park Workshop on Data Abstraction, Database, and Conceptual Modelling [17] marked a figurative and literal high point in expectations for the exchange between databases, programming languages, and artificial intelligence (AI) on conceptual modelling. Quietly hiking amongst the AI, database, and programming language luminaries such as Ted Codd, Mike Stonebraker, Mary Shaw, Stephen Zilles, Pat Hayes, Bob Balzer, and Peter Deutsch was John Mylopoulos, a luminary himself who inspired and guided, for over three decades, the branch of Conceptual Modelling chronicled in this paper.

John’s thoughtful and insightful path started in AI, reached into databases and programming languages, and then into its more natural home at the time, software engineering. In this chapter, I chronicle the leadership provided by John to the field of conceptual modeling. I argue that the conceptual modeling work started thirty years ago under John’s guidance has had far reaching impact on the research in software engineering as well as to its practice in the industry. Conceptual modeling will bloom within the next decade in the form of higher-level programming across all of Computer Science.

2 A Golden Age in Computer Science

The 1970’s and 1980’s marked a golden age of innovation and invention in programming languages, databases, and AI. In programming languages, most of the fundamental programming paradigms used today were invented in that period, including structured [4], object-oriented, and logic programming, as well as the C family. Similarly in databases, the fundamental data models used today were invented then. The Relational Data Model[3] and the Entity Relationship (ER) model[7] transformed databases and data modelling from hierarchical and linked data structures to higher level data abstractions that in turn sparked innovation throughout the 1970’s and 1980’s in data abstraction and semantic data models. AI was emerging from the First AI Winter into a spring of innovation. Semantic nets, which had emerged in the 1950’s and 1960’s [1][2] in psychology and language translation, re-emerged as a focus of AI research in the 1980’s leading to knowledge revolution into expert systems, knowledge-based systems, and knowledge engineering. At this one remarkable time the AI, database, and programming language worlds were all in a

A.T. Borgida et al. (Eds.): Mylopoulos Festschrift, LNCS 5600, pp. 1–9, 2009.

golden age of innovation and invention. While modelling was already an interest in each area, this period marked a Golden Age in the emergence of conceptual modelling due to a confluence of several important developments.

3 The Emergence of Data Modelling

The 1970's were a period of major growth in automation of business applications that led to a dramatic growth in data and transaction volumes that has continued ever since consistently exceeding Moore's Law. This growth was spurred by both demand – the need to automate, and supply – the emergence of database technology. This growth drove the need for data management across an increasingly larger range of business applications and the requirement to model increasingly complex business entities and processes. Database technology became one of the most successful technologies and the bedrock of modern business applications. Yet at the time, the physical and logical database concepts were just emerging.

Prior to the 1970's databases managed data stores of data records and focused on optimizing physical aspects of data storage and retrieval. The emergence of databases as a core tool of business applications led to the need to model business entities as real world business entities rather than mere data records. A basic principle of the emerging database discipline was shared data – a database would manage data for multiple applications. Not only must business entities be modeled in terms of business as opposed to storage, the business entities must be understandable by multiple, possibly unanticipated, applications. Another core database principle was persistence, namely that databases would represent business entities over long periods of time. The requirements to represent business entities logically rather than physically so that they could be shared by multiple applications over long periods of time led to a new research area called data modelling and the search for more expressive data models. How should business entities be modeled? How much real world information should be represented? Where should data modelers turn for inspiration?

In the 1970's, the database world advanced data modelling from the physical level to a more logical level. Since the mid1950's data had been modeled in hierarchically linked data records that represented logical parent:child or 1:N relationships to take advantage of the underlying hierarchical data structures used for storage and retrieval. In 1965 the CODASYL model was introduced to represent more complex data N:M relationships amongst records. While this model increased the expressive power of relationships, it retained many physical aspects. In 1969 Ted Codd made a Turing Award level break-through with the Relational Data Model[3] with which entities and relationships were represented as values in tables, eliminating most physical details and raising data modelling to a new, higher logical level. This was rapidly followed by the ER Model [7] that explicitly permits the modelling of business entities and the relationships amongst them. The relational data model rapidly became adopted as the dominant database model used in the vast majority of database in use today and the ER model became the dominant basis for data modelling, however data modelling at the ER level is practiced by less than 20% of the industrial database world[30], for reasons discussed later.

As the 1970's ended we had an explosion of demand for data management and data modelling, a move towards more logical data models, the emergence of data modelling, and the belief by many data modelers that more expressive models were required. So began a multi-disciplinary pursuit of modeling that had seeds at the Department of Computer Science (DCS) at the University of Toronto (UofT).

4 Seeds of Conceptual Modelling at UofT

The Golden Age in Computer Science was well under way at UofT in all areas of computing including AI, databases, and programming languages. John Mylopoulos was building an AI group, Dennis Tsichritzis was building a database group, and Jim Horning led the programming language / software engineering group. All three research groups were amongst the top five research groups in their respective areas. The multi-disciplinary direction at the university was already active in DCS across these groups in sharing courses, students, and ideas, focusing largely on modelling.

This is when John's passion for modelling, his deep insights, and his quiet guidance took root first at DCS and then beyond. From 1975 to 1980 John supervised many PhDs on modelling topics that drew on AI, databases, and programming languages, or more precisely software engineering. Nick Rossopoulos's 1975 thesis defined one of the first semantic data models[5], which was presented at the International Conference on Very Large Databases in its first session that focused on data modeling and that introduced the ER model[7] as well as two other approaches to conceptual modeling [8][9].

Michael Brodie's 1978 thesis applied programming language data types and AI logical expressions to enhance database semantic integrity augmented by AI modelling concepts to enhance the expressiveness of database schemas and databases. Dennis Tsichritzis, John Mylopoulos, and Jim Horning jointly supervised the work with additional guidance from Joachim Schmidt, the creator of Pascal-R, one of the first database programming languages (DB-PL).

Sol Greenspan's 1980 thesis applied techniques from knowledge representation to formalize the semantics of the SADT software engineering modelling methodology. The resulting paper [16] received the 10-year best paper award due to its adoption by the object-oriented community as a basis for formalizing object-oriented analysis that lead to UML.

While John made many more such contributions, such as Taxis[19], a model for information systems design, and others described elsewhere in this book, the seeds that he sewed in the late 1970's led to a wave of multi-disciplinary modelling efforts in the 1980's beyond UofT.

5 Conceptual Modelling in AI, DB, and PL

The programming language community was first to reach out to the database community to investigate the applicability programming language data type and data structures to data modeling in databases[11]. The leading candidate to share with databases was the programming language notion of data abstraction that came out of structured programming [4] and manifested in abstract data types[5] and that led to object-orientation. This interaction contributed to the Smith's aggregation – generalization data model[12][13] and data modelling methods [14] that were widely accepted in the database community.

The success of the DB-PL interactions on data abstraction and the AI-DB-PL work at UofT, inspired by John Mylopoulos, contributed to the belief that AI, database, and programming languages had mutual interests in data types, data abstraction, and data modelling. This led John Mylopoulos, Michael Brodie, and Joachim Schmidt to hold a series of workshops and to initiate a Springer Verlag book series entitled Topics in Information Systems both dedicated to exploring concepts, tools, and techniques for modelling data in AI (knowledge representation), databases (data modelling), and programming languages (data structures/programming).

The Pingree Park Workshop on Data Abstraction, Databases, and Conceptual Modelling was the highlight of the series. Innovative and influential researchers from AI, databases, and programming languages came with high expectations of mutually beneficial results. The workshop provided area overviews focusing on data modelling aspects and initiated multi-disciplinary debates on challenging issues such as data types, constraints, consistency, behavior (process) vs data, and reasoning in information systems. The proceedings were jointly published in SIGART, SIGPLAN, and SIGMOD[17].

The Intervale workshop on data semantics moved beyond Pingree Park Worshop's focus on data types as means of data modelling, data structuring, and knowledge representation, to comparing AI, database, and programming language models and methods for addressing data semantics in information systems. The results of the workshop[18] were more applicable to and had a greater impact in the AI and databases than they did in programming languages. Logic programming and datalog were introduced in this discussion and was pursued in a later, related, and similarly multi-disciplinary workshop[21].

The Islamorada Workshop Large Scale Knowledge Base and Reasoning Systems[20] extended the Pingree discussion on data types, and the Intervale discussion on modelling data semantics to conceptual modelling in the large - comparing AI knowledge base management systems with database systems[23] and addressed modelling and reasoning in large scale systems.

6 The Contributions of Early Conceptual Modelling

The conceptual modelling workshops and book series contributed to developments in all three areas: semantic data models in databases; object-orientation and UML in programming languages; and knowledge representations such as description logics in AI. Yet, as we will see later, conceptual modelling had a more natural home in software engineering, where John Mylopoulos had sewn conceptual modelling seeds that flourished for two decades. But again more on that later.

While attempts were made in the database community to investigate the potential of abstract data types[15] for modelling and correctness in databases, data types and abstract data types did not gain a footing in database management systems. The DB-PL research domain continued with the annual International Workshop on Database Programming Languages continuing to this day. Similarly databases and database abstractions did not gain a foothold in programming languages. One measure of the successful adoption of a technology is whether the technology is crosses the chasm[25], i.e., adopted by more than the "innovators" and early adopters who constitute less than 15% of the relevant market. In fact, to this day even ER modelling is not widespread in industrial database design [30].

There was a resurgence of interests in abstract data types, and data types in databases in the late 1980's that led to object-oriented databases. The debate that ensued [24] argued the challenges of implementing and using object-orientation in database systems based on the history of relational database management systems. Object-oriented databases died as a research area, but some aspects of objects were incorporated into the object-relational model. IBM made a large investment to incorporate object-relational characteristics into their flagship DBMS, DB2. The systems work required to modify DB2 was enormous and few DB2 customers ever used the object-relational features, just as predicted [24].

A Holy Grail of computing is higher level programming to provide humans with models that are more relevant to the problem domain at hand and to raise the level of modelling and interaction so that, to use IBM's famous motto, people can think and computers can work; and as Ted Codd said for the relational model, to provide greater scope for database and systems optimization by the database management system. So why would conceptual modelling not be adopted by the database world?

My experience with over 1,500 DBAs in a large enterprise and in the broader world of enterprise databases suggests a clear answer. Database design constitutes less than 1% of the database life cycle. Databases tend to be designed over a period of months and then operated for years, sometimes 30, or 40 years. ER modelling is used by a small percentage of practical database designers largely as a tool for analysis and documentation. Once the database design is approved it is compiled into relational tables. Thereafter there is no connection between the ER-based design and the relational tables. During essentially the full life of the database, DBAs must deal with the tables. Databases evolve rapidly in industry. Hence, soon after the database is compiled it is enhanced at the table level and is no longer consistent with the original ER design, had there been one. If, however, the relational tables were kept exactly in sync with the higher-level model so that any changes to one was reflect equivalently in the other, often called "round-trip engineering", the story would be much different. There are additional reasons why conceptual models have not been adopted in the mainstream database industry. The world of Telecommunications billing is extremely complex with literally thousands of features, regulatory rules, banking and credit rules, telecommunications services, choices, and packages. Not only do these features change on a daily basis, the nature of the telecommunications industry and technology leads to fundamental changes in the billing for services. Billing databases are enormous, live for decades, and contain a telecommunication organization's crown jewels. Large telecommunication organizations have many billing systems (hundreds is not uncommon) that must be integrated to provide integrated billing. And there are 1,000s of Telcos. A similar story could be told in ten other areas of telecommunications and in virtually every other industry. ER or conceptual models simply do not (yet) address these large-scale, industrial modelling issues, and if they did, their lack of round-trip engineering would significantly limit their utility.

A recurring lesson in computer science, that has been reinforced in conceptual modeling, is one that originated in philosophy and was adopted by psychology (associative memory), and later language translation[1][2] and reasoning [10] – namely that knowledge can probably be represented using nodes and links or semantic nets as they were originally called in AI. The conceptual modelling work surveyed above has contributed to the development of the node-link model in several areas. While possibly not motivated by those roots, the database world produced many node-link-based conceptual and semantic data models, the most predominant being Chen's ER model[7]. Chen's model has been the most widely adopted to date probably due to its simplicity and tractability. Yet the ER model lacks the expressive power to address the above modeling challenges of the data in telecommunication organizations. A far more expressive node-link-based model is description logics from AI, yet it poses usability issues for industrial database designers. Another area of resurgence of node-link-based knowledge representation is the semantic web. While the first stage of the semantic web was dominated by complex ontologies[28], there is a movement to adopt a far simpler model for augmenting Web resources with meta data, called the Open Links Data[29], which Tim Berners-Lee, the inventor of the web and the co-inventor of the Semantic Web, views as the future direction of the semantic web. The lesson here is not so only the recurrence of the node-link-based model, but also that "A little semantics goes a long way."[1]

7 Conceptual Modelling in Software Engineering and Beyond

The enduring discussion on data and process modelling that started in the 1970's was really between the database and the AI communities[22], sparked and nurtured by John Mylopoulos. John had a deep understanding of the conceptual modeling challenges and opportunities as well as a catholic knowledge of computer science. While his background was in AI, he also understood programming languages from his studies at Princeton, and was present at the birth of relational databases in the 1970's. For the decades from 1979 to 2009 John was key to most of the developments in conceptual modelling either directly as a contributor or indirectly as a mentor and connector across communities – across AI, database, and software engineering communities, and across various AI factions, for example, Europe vs. North America or description logics vs. datalog.

John also realized that it was the software engineering community that focused on the initial design and modelling stage of the database life cycle. It is also concerned with logical and physical requirements, specification of integrity and data quality, and the evolution of data, process and other models. Indeed, data modelling is now considered a software engineering activity rather than a database activity, as data modelling is an integral component with process modelling in the information systems life cycle.

John pursued conceptual modelling as a software engineering activity in the early 1980's when he supervised PhD theses[16][19] that contributed to mainstream

[1] Profound and now famous remark by Jim Hendler, Tetherless World Chair of Computer Science, Rensselaer Polytechnic Institute.
http://www.cs.rpi.edu/~hendler/LittleSemanticsWeb.html

software engineering such as the languages and methods surrounding UML. Hence, the software engineering community became the beneficiary of conceptual modelling and extended it to address software engineering issues, discussed elsewhere in this volume.

Now the story gets better as John Mylopoulos probably realized long ago. To return to the programming Holy Grail, humans should use high-level representations that permit them to understand the system in logical, human terms as opposed to machine level terms. Higher-level representations enable better design, analysis, monitoring, adaptation, and manipulation. Not only are higher-level representations more understandable by humans, they are also less error prone and lead to considerably higher productivity.

The challenge in achieving higher-level programming is to map the higher-level representations onto machine level representations precisely (i.e., the same semantics), equivalently (modifications in one map to semantically equivalent changes in the other), and in ways that are optimal and scalable as the system evolves in capability and grows in data and transaction volumes.

In 2001 OMG launched the Model-driven architecture (MDA) initiative to strive towards this long sought after Holy Grail. MDA is a software engineering approach for information systems development that uses models to guide systems architecture design and implementation with the objective of developing and maintaining a direct connection between the high-level model and the executable representations, to achieve the desired round-trip engineering. But you need more than a direct connection, i.e., equivalence between the high- and low-level models. Information systems evolve rapidly. Hence, changes to the high-level model, required to meet changing logical requirements must be reflected in the low-level model and changes in the low level model for optimization must be reflected equivalently in the high-level model. This capability is called agile or adaptive software development.

MDA and agile software development objectives are becoming adopted in industry with projections that initial results will be ready for industrial use in 2012. For example, Microsoft has announced support of MDA by Oslo[30] that is a forthcoming model-driven application platform.

Once MDA and agile software development are in industrial use, the entire systems life cycle can operate simultaneously at two levels – high-level models for human understanding, analysis, and modification and the executable level. This will address the lack of round trip engineering that limit the utility of today's modelling systems. At that point the modelling concepts that initiated with Conceptual Modelling, many inspired directly or indirectly by John Mylopoulos, will be directly usable across the life cycle and computing will move to a higher level – to domain models such as Telecom billing and airline reservations - and these models will be constructed with concepts, tools, and techniques that evolved from the seeds sewn in the Golden Age on computing. This will bring models and modelling to a more professional level in which models are developed by modelling experts and are standardized for reuse in the respective industry to address many of today's integration and semantic challenges.

8 And Beyond That

For three decades John Mylopoulos quietly inspired generations of researchers in the ways of conceptual modelling from his base at the University of Toronto. His vision, persistence, and insight quietly directed theses, researchers, and indeed the conceptual modelling area with contributions to AI, databases, programming languages, and software engineering. This too will get better. John is continuing his path from a new base in the mountains of Trento, Italy and soon the results of his efforts – the flowers from the seeds sewn over the three decades – will be accessible to all of computing and modelling itself will become a professional domain with John as one of its major contributors.

I am grateful for John's friendship, wisdom, and his quiet way of being – open and willing to talk and inspire us all.

References

[1] Collins, A.M., Quillian, M.R.: Retrieval time from semantic memory. Journal of verbal learning and verbal behavior 8(2), 240–248 (1969)

[2] Collins, A.M., Quillian, M.R.: Does category size affect categorization time? Journal of verbal learning and verbal behavior 9(4), 432–438 (1970)

[3] Codd, E.F.: A Relational Model of Data for Large Shared Data Banks. Commun. ACM 13(6), 377–387 (1970)

[4] Dahl, O.-J., Dijkstra, E.W., Hoare, C.A.R.: Structured Programming. Academic Press, London (1972)

[5] Liskov, B., Zilles, S.N.: Programming with Abstract Data Types. SIGPLAN Notices (SIGPLAN) 9(4), 50–59 (1974)

[6] Roussopoulos, N., Mylopoulos, J.: Using Semantic Networks for Database Management. In: VLDB 1975, pp. 144–172 (1975)

[7] Chen, P.P.: The Entity-Relationship Model: Toward a Unified View of Data. In: VLDB 1975, p. 173 (1975)

[8] Navathe, S.B., Fry, J.P.: Restructuring for Large Data Bases: Three Levels of Abstraction. In: VLDB 1975, p. 174 (1975)

[9] Senko, M.E.: Specification of Stored Data Structures and Desired Output Results in DIAM II with FORAL. In: VLDB 1975, pp. 557–571 (1975)

[10] Collins, A.M., Loftus, E.F.: A spreading-activation theory of semantic processing. Psychological Review 82(6), 407–428 (1975)

[11] Organic, E.I.: Proceedings of the 1976 Conference on Data, Abstraction, Definition and Structure, SIGPLAN Notices, Salt Lake City, Utah, United States, March 22 - 24, vol. 11(2) (1976)

[12] Smith, J.M., Smith, D.C.P.: Database Abstractions: Aggregation and Generalization. ACM Trans. Database Syst. (TODS) 2(2), 105–133 (1977)

[13] Smith, J.M., Smith, D.C.P.: Database Abstractions: Aggregation. Commun. ACM (CACM) 20(6), 405–413 (1977)

[14] Smith, J.M., Smith, D.C.P.: Principles of Database Conceptual Design. Data Base Design Techniques I, 114–146 (1978)

[15] Brodie, M.L., Schmidt, J.W.: What is the Use of Abstract Data Types? In: VLDB 1978, pp. 140–141 (1978)

[16] Greenspan, S.J., Mylopoulos, J., Borgida, A.: Capturing More World Knowledge in the Requirements Specification. In: ICSE 1982, pp. 225–235 (1980)

[17] Brodie, M.L., Zilles, S.N. (eds.): Proceedings of the Workshop on Data Abstraction, Databases and Conceptual Modelling, Pingree Park, Colorado, June 23-26 (1980); SIGART Newsletter 74 (January 1981), SIGMOD Record 11(2) (February 1981), SIGPLAN Notices 16(1) (January 1981) ISBN 0-89791-031-1

[18] Brodie, M.L., Mylopoulos, J., Schmidt, J.W. (eds.): On Conceptual Modelling, Perspectives from Artificial Intelligence, Databases, and Programming Languages, Book resulting from the Intervale Workshop 1982, Topics in Information Systems. Springer, Heidelberg (1984)

[19] Mylopoulos, J., Borgida, A., Greenspan, S.J., Wong, H.K.T.: Information System Design at the Conceptual Level - The Taxis Project. IEEE Database Eng. Bull. 7(4), 4–9 (1984)

[20] Brodie, M.L., Mylopoulos, J. (eds.): On Knowledge Base Management Systems: Integrating Artificial Intelligence and Database Technologies, Book resulting from the Islamorada Workshop 1985, Topics in Information Systems. Springer, Heidelberg (1986)

[21] Schmidt, J.W., Thanos, C. (eds.): Foundations of Knowledge Base Management: Contributions from Logic, Databases, and Artificial Intelligence, Book resulting from the Xania Workshop 1985. Topics in Information Systems. Springer, Heidelberg (1989)

[22] Brodie, M.L., Mylopoulos, J. (eds.): Readings in Artificial Intelligence and Databases. Morgan Kaufmann, San Mateo (1989)

[23] Brodie, M., Mylopoulos, J.: Knowledge Bases and Databases: Current Trends and Future Directions. In: Karagiannis, D. (ed.) IS/KI 1990 and KI-WS 1990. LNCS, vol. 474. Springer, Heidelberg (1991)

[24] Stonebraker, M., Rowe, L.A., Lindsay, B., Gray, J., Carey, M., Brodie, M., Bernstein, P., Beech, D.: Third Generation Data Base System Manifesto" (with). ACM SIGMOD Record 19(3) (September 1990)

[25] Crossing the Chasm: Marketing and Selling High-tech Products to Mainstream Customers (1991, revised 1999) ISBN 0-06-051712-3

[26] Stonebraker, M., Moore, D.: Object-Relational DBMSs: The Next Great Wave. Morgan Kaufmann, San Francisco (1996)

[27] The history of conceptual modeling, `http://cs-exhibitions.uni-klu.ac.at/index.php?id=185`

[28] OWL Web Ontology Language Reference, W3C Recommendation (February 10, 2004), `http://www.w3.org/TR/owl-ref/`

[29] Berners-Lee, T., et al.: Linked Data: Principles and State of the Art, keynote. In: 17th International World Wide Web Conference, Beijing, China, April 23-24 (2008)

[30] Hammond, J.S., Yuhanna, N., Gilpin, M., D'Silva, D.: Market Overview: Enterprise Data Modeling: A Steady State Market Prepares to Enter A transformational New Phase. In: Forrester Research, October 17 (2008)