Philosophical Foundations of Conceptual Modelling: What is a Conceptual Model?

John Mylopoulos
University of Ottawa

ER Online Summer Seminars
Free University of Bozen-Bolzano
July 1, 2020

Acknowledgements

- This presentation draws on joint work with Nicola Guarino and Giancarlo Guizzardi [Guarino19].
- My work on Conceptual Modelling and Knowledge Representation since the early 1970s was greatly influenced by the ideas of Alex Borgida (RutgersU, emeritus) and Hector Levesque (UToronto, emeritus).

Abstract

We review and analyze three theses to a fundamental question for Conceptual Modelling, “What is a conceptual model?” For each thesis we offer evidence that it is credible and consider some of the research questions it entails. Our study draws ideas from Philosophy, Cognitive Science, Engineering and the Social Sciences, as well as several areas within Computer Science, including Databases, Software Engineering (SE), Artificial Intelligence (AI) and Information Systems Engineering (ISE).

Context

- A recent ER paper [Delcambre18] proposes “A Reference Framework for Conceptual Modelling” that attempts to answer fundamental questions about Conceptual Modelling, such as: What is and isn’t a conceptual model? Who does conceptual modelling? etc.
- The paper includes the results of a workshop involving 40 or so experts who discussed alternative answers.
- The results suggest a surprising diversity of views.
- In this presentation, I consider three complementary theses, answers to the question “What is a conceptual model?”, referred to respectively as T.I, T.II and T.III.

Why Philosophical Foundations?

- Because the answers we give to fundamental question about any research area tell us what is relevant research we should take into account; they also suggest research problems to tackle, and methodological tools to use.

T.I Conceptual models as models of mental representations

- It is a basic tenet of Cognitive Science and Philosophy of Mind that cognitive processes create, use and transform ***mental representations*** of the world [SEP05].
- These can be conceptual (consisting of concepts), e.g. thoughts, or non-conceptual, e.g. sensations.
- ***Conceptual models*** are ***computational*** models of conceptual mental representations, to be used for purposes of understanding, communication and problem solving.
- Conceptual models are unique among models used traditionally in Science and Engineering in that they don't model the world, but rather ***our conceptualizations of the world***.

Computationality

- Conceptual models are computational in that they are stored in computers and are analyzed and reasoned with through computational means.
- Computationality renders conceptual models scalable and analyzable. Pragmatically speaking, it is inconceivable that conceptual modelling would be viable or interesting to potential users if they were not computational.
- It is no accident that conceptual models came about after the advent of computers.
- Computationality implies that Conceptual Modelling as a research area covers algorithmic issues (e.g., complexity of reasoning mechanisms) and database issues (e.g., scalable query processing for large conceptual models).

T.I raises research questions

- What do we know about the properties and structure of conceptual mental representations?
- What modelling languages are appropriate for representing conceptual mental representations?
- How do we ensure that a conceptual model represents faithfully a particular mental representation?
- Note that mental representations are not observable, so far: nobody has ever seen or otherwise sensed these phantom mental phenomena.

Concepts as atoms of thought

- Concepts are the atoms of conceptual mental representations. They are formed through experience (Locke, Hume) and provide a “lens for looking at the world”, or “a language of thought” [Fodor75].
- There seems to be consensus that concepts come with a definitional structure, are associated to other concepts (Associationism, [SEP18]), and have instances/referents.
- Most conceptual modelling languages adopt this associationist perspective: semantic networks, semantic data models, description logics, OO models, …
- Others adopt a logical perspective: Concepts have a propositional, rather than an associationist, structure.

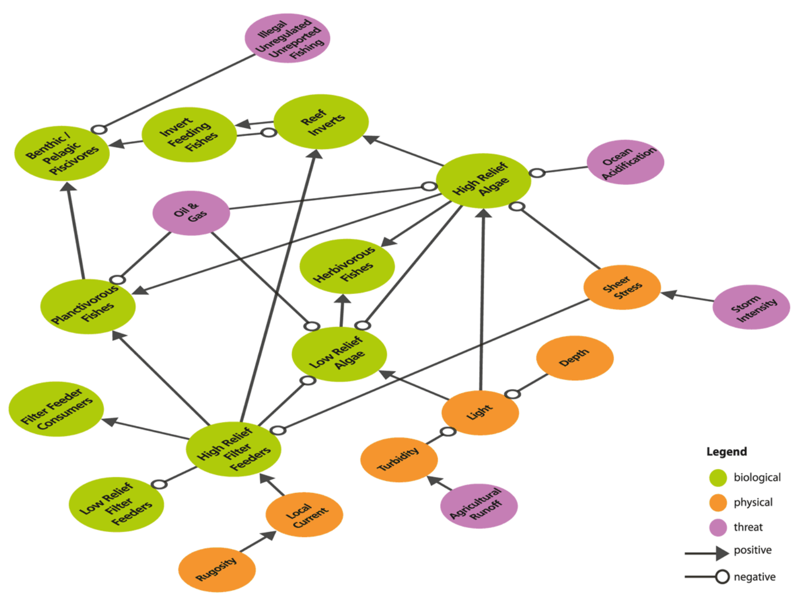
Qualitative models

- AI has studied ***qualitative models*** that model a domain, but in qualitative terms [Kuipers94].
- For example, the model of a house may include rooms labelled large/medium/small, rather than specify their size; Such models are used for qualitative reasoning, a form of common sense reasoning.
- But such models model a ***domain*** in qualitative terms, not somebody's conceptualization of that domain.
- For instance, we may use 'qualification rules' such as R to build a qualitative model of a building from a quantitative one.

  R := rooms of size ≤9sqm are labelled 'small', between 9sqm and 64sqm 'medium' and over 64sqm 'large'

Qualitative models (con'd)

- Note that two qualitative models of the same domain will always be the same if they use the same qualification rules; in that sense they are ***objective***.
- But they need not be the same if they are conceptual; conceptual models are intrinsically ***subjective***.



… But wait …

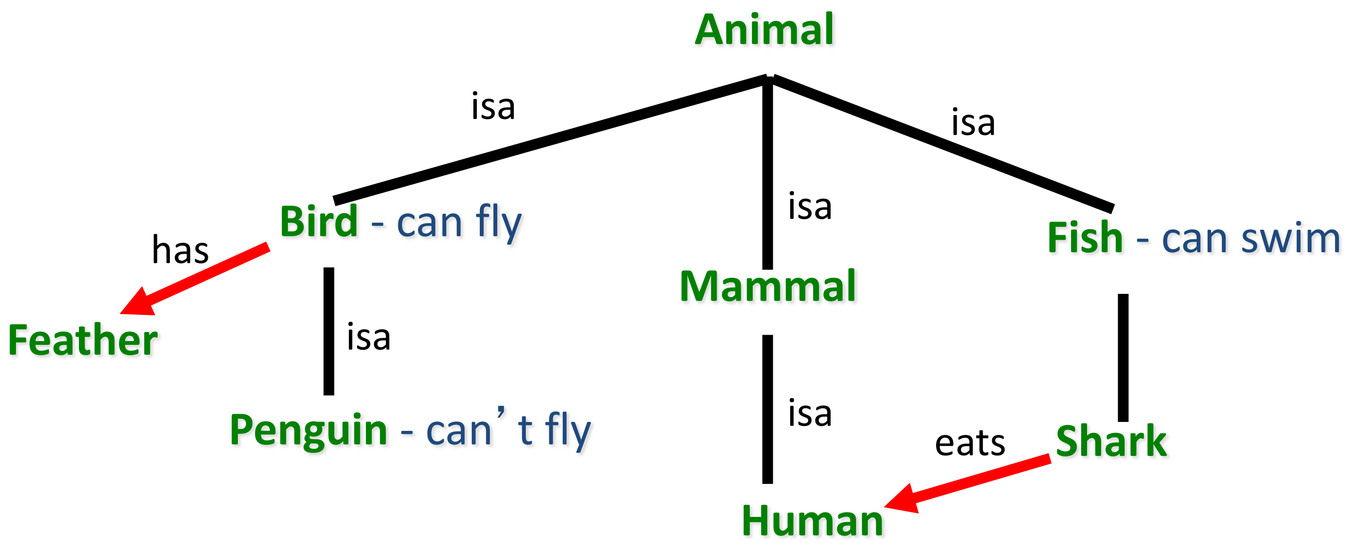
What evidence do we have in support of the thesis that conceptual models are models of mental representations?

Brief history of Conceptual Modelling

- In AI, Ross Quillian proposed *semantic networks* as a model of human memory [Quillian66].
- In Programming Languages, Ole-Johan Dahl proposed *Simula* for simulation programs [Dahl68].
- In Databases, Jean-Raymond Abrial proposed a *semantic model* in 1974 [Abrial74], shortly followed by Peter Chen's *entity-relationship model* [Chen76] for modelling the contents of a database
- Doug Ross proposed the *Structured Analysis and Design Technique* (*SADT*) as a "language for communicating ideas" [Ross77] and used it to define software requirements

[Mylopoulos98]

Semantic networks (Quillian 1966)



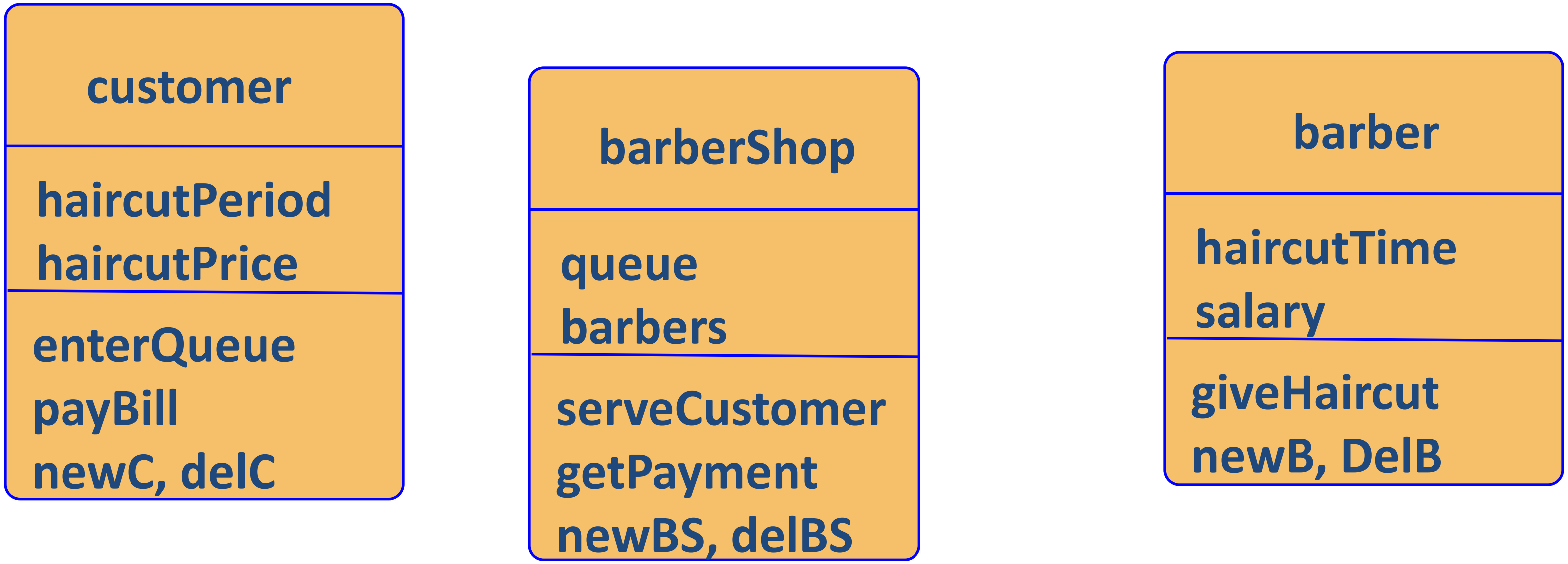
Novel ideas

- Models are built out of *concepts*, *associations* and *attributes*
- Attributes and associations are inherited by default, from a single concept
- Computation is defined in terms of *spreading activation* -- e.g., discovering different interpretations of “horse food”

  horse --> animal –eats-> food

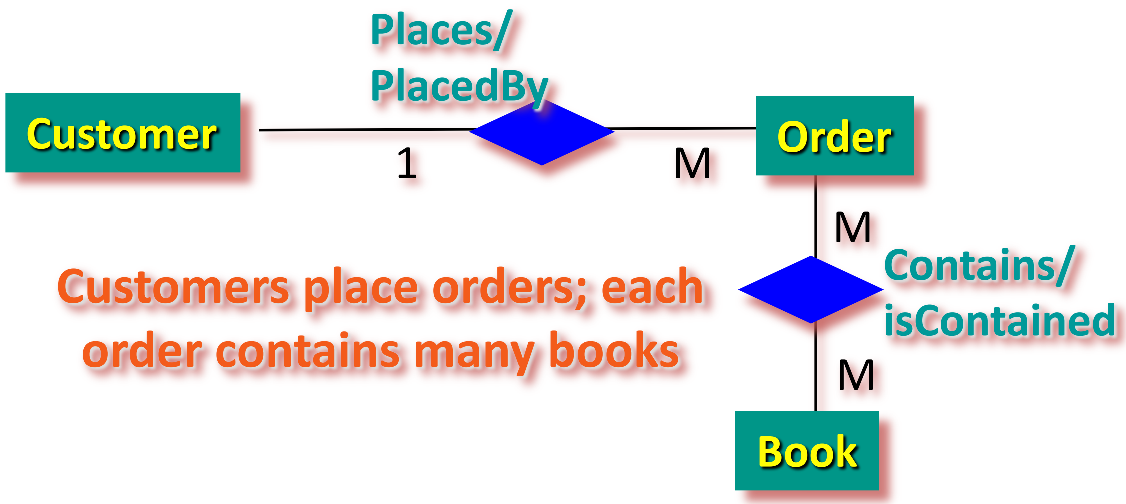
  horse --> animal –madeOf-> meat --> food

Simula (Dahl 1967)



- Proposed as an extension of ALGOL 60, to build easy-to-understand models of a domain for simulation purposes.
- A (simulation) program consists of classes and instances.
- Classes model concepts, their properties and behaviours, are organized into subclass hierarchies.
- Their instances model objects in the domain.

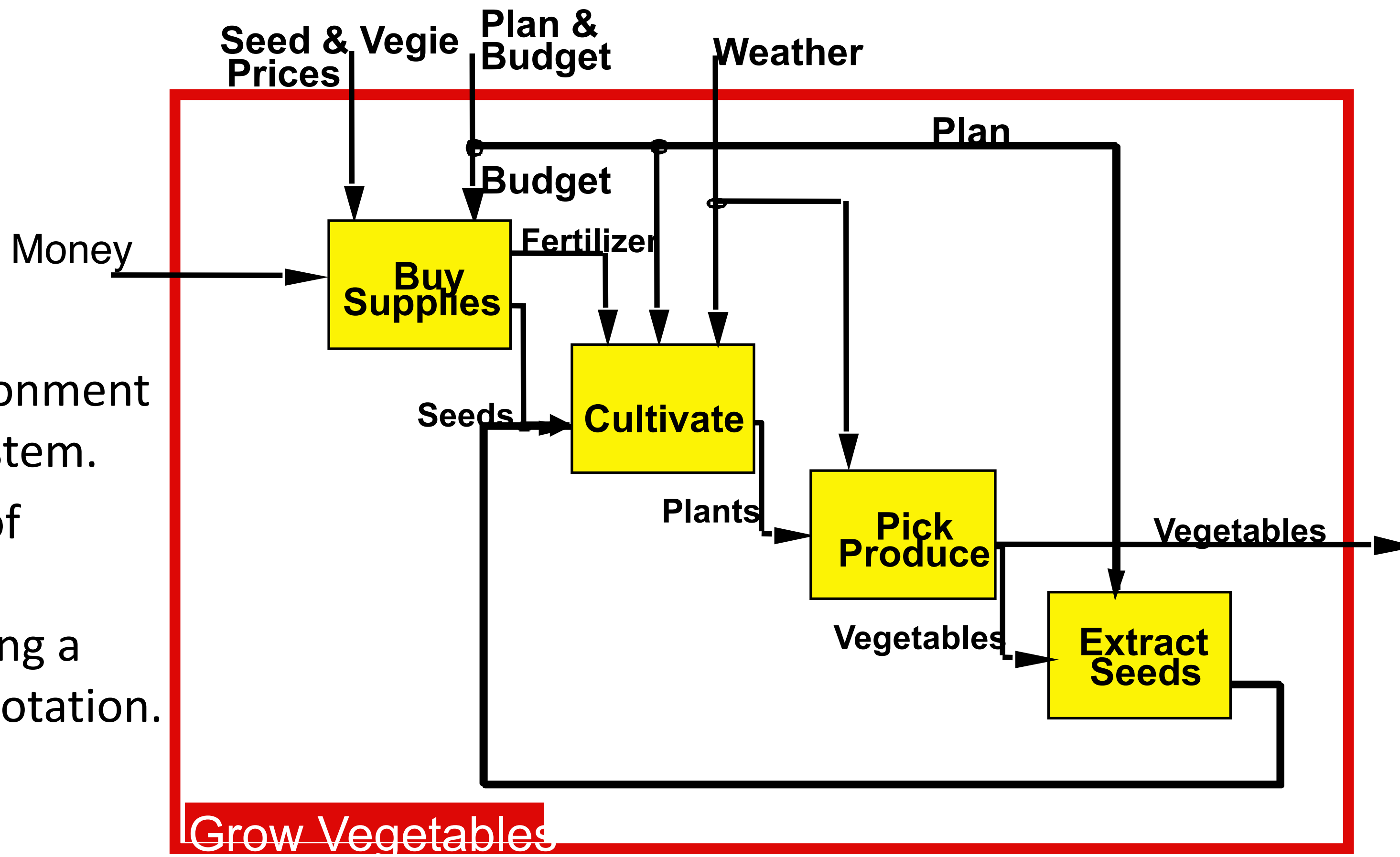
Entity-Relationship Model (Chen 1975)



Novel ideas

- Assumes that the domain of a database consists of *entities* and *relationships* (*ontological assumption*)
- Shows how a conceptual schema can be mapped onto a logical one.
- [Abrial's semantic model was more akin to OO data models, but did offer entities and relations too as modelling primitives]

SADT -- Structured Analysis and Design Technique (Ross 1977)



Novel Ideas

- Model the environment of a software system.
- Model consists of ***activities*** and ***data***/***objects*** using a box-and-arrow notation.

Brief history (cont'd)

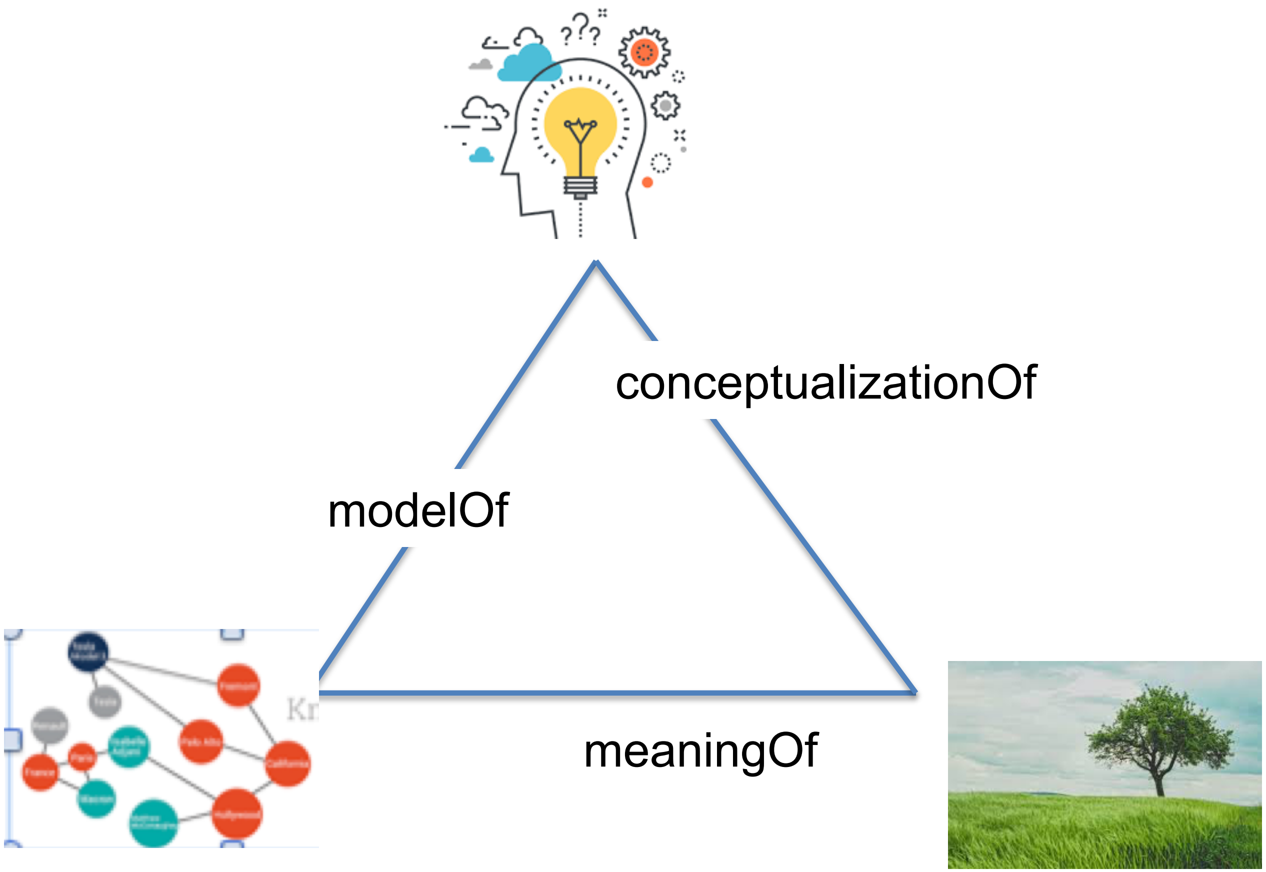
- In AI, semantic networks served as foundation for knowledge representation languages, e.g., KLONE [Brachman78] and PSN [Levesque79], leading to Description Logics [Baader03] and OWL, a WWW standard.
- Dahl's Simula evolved into Smalltalk at Xerox PARC, led to OO Modelling and UML, also the Models conferences.
- The ER model led to the ER conference series, and is used routinely in industrial practice for conceptual schema design.
- SADT was followed by formal requirements languages, [Greenspan82], goal-oriented ones, KAOS [Dardenne93], i* [Yu97] and led to the RE conference series.
- New research threads of Conceptual Modelling have sprang in Business Process Management (BPM), Enterprise Architectures (EAs) and more.

Common theme

- A common theme that underlies all these efforts is the idea of building models that view a domain “the way we do” and are consequently easier to conceive, build and understand; they are “intuitive”, “direct and natural” representations [Hammer78] they “capture true meaning” [Roussopoulos75].
- This common theme suggests that early proposals for conceptual models were meant to capture (aka model) our conceptualizations of a domain, rather than the real thing.
- Note that concepts and their relationships don’t exist in any domain, only in someone’s conceptualization of that domain.

“Direct and natural”

- “***Direct and natural***” means that they have a semantics relative to the domain (direct), and they reflect our conceptualizations of that domain (natural).



Uses of conceptual models

- In AI and Knowledge Representation (KR) these models serve as ***knowledge bases*** for artificial agents who conduct problem solving.
- In Databases, SE, BPM, EA they serve as ***design models*** for databases, software, business processes, enterprises, useful for understanding and communication.

How do we know anything about our conceptualizations?

- From ***things we say*** about our conceptual mental representations.
- For example, if you ask a database administrator to describe for you the contents of her database and she tells you that it's about students taking courses, having marks, etc., you can come up with an ER schema for it*.
- In fact, Databases textbooks do teach how to go from a natural language description of the contents of a database to a conceptual schema, see [Atzeni99].

____

* But not a relational schema, because ***that*** is not a conceptual model.

Models in Science and Engineering

- They are usually models of the spatio-temporal properties of physical things.
- They are ***objective*** in that they model the domain, not someone's conceptualization of the domain. Moreover, they don't include the concepts they use.
- Consider the model of a building: Engineers use measuring instruments to construct a 3D model; to build a conceptual model, you'd ask an occupant to describe the building for you.

Newton's second law, conceptually and otherwise

- The law states that F = M * A, where F stands for 'force', M for 'mass' and A for 'acceleration'.
- But where are the concepts? Obviously, M and A are qualities of a ***physical object***, while F represents the strength of an ***action*** (push) applied to that object, and the law is relating the strength of the applied force to the object's physical properties M and A.
- A conceptual model of this law would include entity class PhysObject with attributes mass, velocity and acceleration, action class ApplyForce with object and strength attributes.

Corollary: A completeness criterion for conceptual models

- Since conceptual models are to be used for understanding and communication, they must satisfy two properties:
  - ✓ Every instance must be associated to relevant classes it is an instance of;
  - ✓ Every attribute must be associated to its subject.
- Conceptual models are actually used in Physics for pedagogical purposes [MacKay19], but not for defining physical laws …

Conceptual models for *some* Sciences and Engineering

- Conceptual models capture ***subjective*** views of a domain.
- Some Sciences need such models, including Cognitive Science, Social Sciences, Law, Economics, Management Sciences, and Computer Science.
- Same for Engineering: Much of Engineering has to do with the design of artifacts [Simon69]. And conceptual models have proven themselves essential tools for doing design.

Many models in Computer Science are not conceptual

- Machine Learning models are statistical, there are no concepts.
- Programs are not conceptual either, they consist of instances of the concepts of procedure, data type, variable, etc., but don't include the concepts themselves.
- Relational schemas are not conceptual as well, because they say nothing about the meaning of data in a database, only about its structure.

Concepts must have coherence

- Empiricists argue that concepts are learned from experience.
- Most conceptual modelling languages represent concepts as predicates and use connectives to define new concepts.
- For example, say Man and Unmarried are modelled as predicates $\mathbb{M}$ and $\mathbb{U}$, then define Bachelor as $\mathbb{M} \wedge \mathbb{U}$.
- According to this view, $\neg\mathbb{M}$ and $\mathbb{M} \vee \mathbb{N}$ define concepts too. But do they? Concepts are supposed to capture ***observable shared properties*** of their instances ('coherence').
- Concepts such as ANY (with everything as an instance), and ¬C and C1 ∨ C2 are problematic because they have no coherence; hence they are not learnable.

Concepts as prototypes

- Concepts in mainstream conceptual modelling languages are defined in terms of necessary and sufficient rules and instantiation is all-or-nothing. This leaves no room for relating a cat-like figure to the concept of cat.
- In prototype theory, on the other hand, instances of a concept share a "family resemblance" [Wittgenstein53]. This theory seems to offer a better account for what we know about conceptual mental representations.
- There has been work in KR that adopts a prototype perspective for concepts [Minsky74].
- There are other theories as well about the structure of concepts [SEP05].

T.II Conceptual models as artifacts

- It is a basic tenet of Engineering that artifacts are designed to fulfill their ***requirements***.
- Conceptual models are undoubtedly artifacts so they should have requirements, Engineering dictates.
- Moreover, according to Engineering, the quality of an artifact is determined by how well it fulfills its requirements (quality as "fitness-to-purpose").
- It is interesting that requirements for conceptual models is a topic that has drawn very little attention within the ER community.

Requirements for conceptual models

- For software, requirements concern the ***functions*** and ***qualities*** that the system-to-be ought to have.
- For conceptual models, requirements concern the ***coverage*** and the ***qualities*** that the model-to-be ought to have.
- For example, an enterprise model for a university department, may be required to cover strategic objectives, academic programmes, research activities, and academic staff; for qualities we may want it to be 100% complete (no false negatives) and ≥80% sound (few false positives).
- Who are the stakeholders? Members of the department with different areas of expertise, also modellers.

T.III Conceptual models as social artifacts

- A conceptual model is shaped by the perspective of the cognitive agent whose mental representations it captures.
- But, that agent need not be an individual, it could be a social one, a group or an organizational unit, whose members are the ‘stakeholders’.
- Now the conceptual model is a social artifact in that it needs to capture the shared conceptualization of a group.
- In fact, the conceptual models used for design, be they in Databases, Software Engineering, Business Process Management, etc. embrace this social perspective.
- (BIG!) Problem: the shared conceptualization often doesn’t exist, only the individual conceptualizations of its members.

Social conceptual models

- The modelling process for such models has to produce a shared conceptualization ***and*** build a model for it.
- Such a process has to be dialectic with stakeholders as participants, that starts with an individual's conceptual model and evolves it into a shared one through an iterative process of thesis-antithesis-synthesis.
- For the University department example, modelling may start with the perspective of the department chair, enriched through argument with those of others.
- Dialectics has an illustrious history; its current incarnation in Computer Science is Argumentation Theory [Dung95].
- [Borgida20] adopts such a dialectic perspective for building requirements models.

Implications

- T.I adds a scientific perspective to Conceptual Modelling: we are building models of ***mental phenomena*** that no one has ever observed, but are nevertheless accepted as things that exist and actually play a critical role in cognition.
- T.II draws from fundamental principles in Engineering to posit the necessity for adding a new component to any conceptual modelling project: the definition, analysis and validation of model ***requirements***.
- T.III draws ideas from Dialectics in Philosophy and the Social Sciences to posit conceptual modelling as a ***dialectic process*** of thesis-antithesis-synthesis.

Conclusions

- Conceptual models were initially proposed in several areas of Computer Science as models of our conceptualizations of a domain.
- Conceptual Modelling is, or should be, a core topic in Computer Science, to be taught at the undergraduate and graduate level.
- Equally importantly, Conceptual Modelling constitutes one of the major contributions of Computer Science to other Disciplines in that it offers a subjective type of modelling for applications that don’t subscribe to the idea of an objective truth.

References

- [Abrial72] Abrial J-R., “Data Semantics”, in Klimbie J. and Koffeman K. (eds) *Data Management Systems*, North-Holland, 1972.
- [Atzeni99] Atzeni P., Ceri S., Paraboschi S., Torlone R., *Database Systems: Concepts, Languages and Architectures*, The McGraw-Hill Companies, 1999.
- [Baader03] Baader F., Calvanese D., McGuiness D., *The Description Logic Handbook: Theory, Implementation, Applications*, Cambridge University Press, 2003.
- [Borgida20] Borgida A., ElRakaiby Y., Ferrari A., Mylopoulos J., “A Refinement Calculus for Requirements Engineering based on Argumentation Theory”, submitted for publication.
- [Brachman77] Brachman, R., “What’s in a Concept: Structural Foundations for Semantic Networks”. *International Journal of Man-Machine Studies 9(2*), 127–152, 1977.
- [Chen76] Chen P., “The Entity-Relationship Model: Towards a Unified View of Data“, *ACM Transactions on Database Systems 1(1)*, 9-36, 1976.
- [Dahl68] Dahl O-J., Myhrhaug B., “Some Features of the Simula 67 Language” 2nd Conference on Applications of Simulations. 29–31, 1968.
- [Dardenne93] Dardenne A., van Lamsweerde A., Fickas S., “ Goal–directed Requirements Acquisition”, *Science of Computer Programming, 20*, 1993.
- [Delcambre18] Delcambre L., Liddle S., Pastor O., Storey V., “A Reference Framework for Conceptual Modeling”, International Conference on Conceptual Modeling, Springer, 2018.
- [Dung95] Dung, P.M.: On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. AI Journal 77(2), 321–357 (1995)

References (cont'd)

- [Fodor75] Fodor, J. (1975). *The Language of Thought,* Cambridge, MA: Harvard University Press.
- [Greenspan82] Greenspan S., Mylopoulos J., Borgida A., "Capturing More World Knowledge in the Requirements Specification", 6th International Conference on Software Engineering, Kyoto, 1982.
- [Guarino19] Guarino N., Guizzardi G., Mylopoulos J., "On the Philosophical Foundations of Conceptual Models", 29th European-Japanese Conference (EJC'19), Lappenranta Finland, June 2019.
- [Hammer78] Hammer M., McLeod D., "The Semantic Data Model: A Modelling Mechanism for Database Applications", ACM SIGMOD Conference, 1978.
- [Kuipers94] Kuipers B., Modelling and Simulation with Incomplete Knowledge, MIT Press, 1994.
- [Levesque79] Levesque H., Mylopoulos J., "A Procedural Semantics for Semantic Networks". in *Associative Networks: Representation and Use of Knowledge by Computers,* 93–120, 1979.
- [MacKay19] serc.carleton.edu/sp/library/conceptualmodels/index.html.
- [Minsky74] Minsky M., "A Framework for Representing Knowledge", memo no.306, AI Lab, MIT, 1974.
- [Mylopoulos98] Mylopoulos J., "Information Modeling in the Time of the Revolution"*, Information Systems 23(3-4*), 1998.
- [Page18] Page S.*, The Model Thinker*, Basic Books, 2018.
- [Quillian68] Quillian R., "Semantic Memory", in Minsky M. (ed.) *Semantic Information Processing*, MIT Press,1968.
- [Ross77] Ross D., "Structured Analysis (SA): A Language for Communicating Ideas". *IEEE Transactions on Software Engineering 3(1)*, 16–34, 1977.

References (cont'd)

- [Roussopoulos75] Roussopoulos N and Mylopoulos J., "Using Semantic Networks for Database Management", First Very Large Databases Conference, Austin, 1975.
- [SEP05] "Concepts", *Stanford Encyclopedia of Philosophy*, 2005.
- [SEP18] "Associationist Theories of Thought", *Stanford Encyclopedia of Philosophy*, 2018.
- [Simon69] Simon H., *The Sciences of the Artificial*, MIT Press 1969.
- [Yu97] Yu E., "Towards Modelling and Reasoning Support for Early Phase Requirements Engineering", 3rd IEEE International Symposium on Requirements Engineering, Annapolis, January 1997.
- [Wittgenstein53] Wittgenstei, L., *Philosophical Investigations*, Oxford Blackwell, 3rd edition, 1953.