DOI: 10.1145/1516046.1516079

BY MARK LYCETT AND CHRIS PARTRIDGE

The Challenge of Epistemic Divergence in IS Development

THE ORGANIZATIONAL ENVIRONMENT INCREASINGLY demands that computer-based information systems are responsive to change and can work with each other seamlessly (ideally from a dynamic perspective). Given the large investment that organizations have in mission-critical legacy systems, evolutionary maintenance and systems integration now form a very significant part of the cost and effort profile of systems development. In terms of the integration issue, much of the difficulty lies in the fact that different systems often contain different 'representations' of the world. In the development process, it is generally accepted that the 'information' an information system contains about its *business domain(s)* is an essential intellectual part of the system, and the domain of fundamental concern. This concern is generally regarded as unitary, however, requiring no further breakdown into parts and it is commonly perceived that its relation to the business information system is simple and direct.

This article questions that assumption, proposing that the link between the 'informational' aspects of an information system with the business domain is based upon a simplification – one which is detrimental to systems development in an age of interoperability. Our observations are drawn from work on re-engineering ontology-based business domain models from commercial legacy systems. Doing this has revealed that the informational aspect of a system has a finer-grained underlying structure driven by the dual concerns of *ontology* and *epistemology*. Ontology deals, as its name suggests, with what actually *is*, the objects that make up the business domain. Epistemology deals with what the system knows about this ontology – this is a key relation between the system and its ontology (business domain). We articulate the issue here through the concept of 'epistemic divergence' which, simply put, is the differences between ontology and epistemology in any given system. While ontology is increasingly employed in systems development, there is little discussion of epistemology in the sense used here – or of its distinction from the ontological concern. Explaining epistemic divergence in terms of these finer-grained concerns undermines the common assumption of a simple referential relation between the business domain and the information system and reveals it as a simplification that takes no account of the fact that the semantics of the information in working systems is also shaped by epistemological concerns.

In exposing the issue of epistemic divergence, we first examine the fundamental assumptions of the mainstream approach to systems development. With those assumptions made clear, we then provide examples of two common forms of epistemic divergence that we have identified during our reengineering activities. Last we use the outcomes of those examples to examine the implications for research and practice in future systems development if we are to account for epistemic divergence.

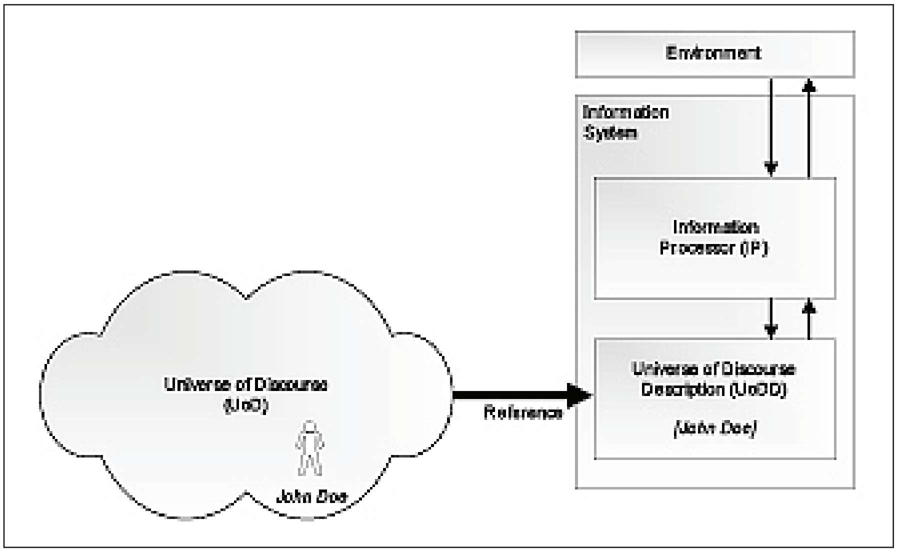
The Mainstream View

Business information systems contain information *about* business things. While academic alternatives exist, the prevailing commonsense (folk) view assumes there is a clear and simple referential relationship between the information in an Information System and the (business domain) objects it describes – regarding this as the way that the business domain shapes the 'information in the system'. The foundational work to formalize this view was done in the late 1970s and early 1980s.[1,3,5] Aside from the referential relationship, commonly accepted influential notions deriving from such work include the data/process distinction and the separation of concerns.

In the mainstream paradigm, an information system is taken to contain information about the real world. As the system is also in the real world, the (inelegant) technical term Universe of Discourse (UoD) was adopted from logic to name that portion of the real world that was being described (the collection of all things in a portion of the real world). The information or description was then called the Universe of Discourse Description (UoDD), and determining the content of the UoD is effectively determining the content of the UoDD. Thus, the UoDD is that part of the information system that is shaped by the UoD – the description of the business domain. The assumption underlying this view is that the information (UoDD) in the information system can be clearly segregated and that its 'about' relation with the UoD is fundamentally referential - which implies that the semantics of the UoDD will be a straightforward reflection of the UoD. The referential aspect is well-documented.[4]

The UoD/UoDD paradigm situates the part of the information system shaped by the business domain (UoDD) into a general framework for information systems. In general terms, the framework is segregated into the description of the business domain (UoDD), actions performed on the description of the business domain (the Information Processor) and the environment (the part of the world with which the information system will interact) as illustrated in Figure 1.

Figure 1. Information System, Environment and Universe of Discourse

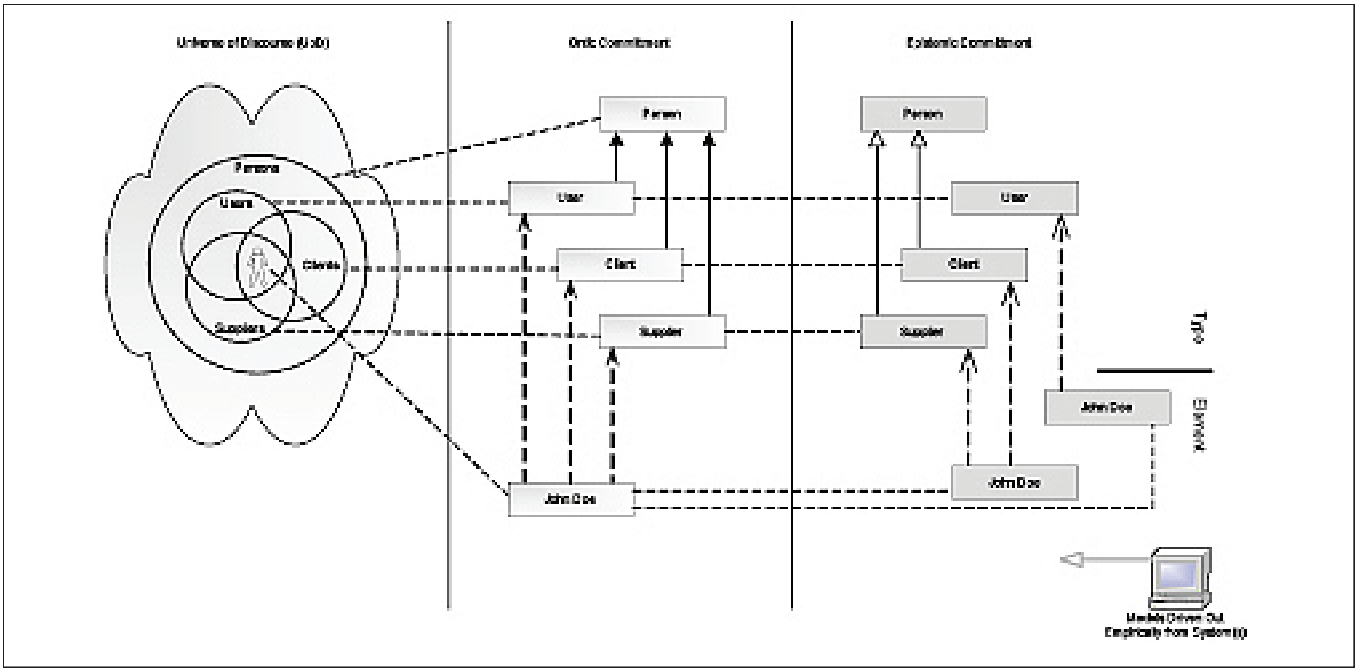

Environment
Information System
Information Processor (IP)
Universe of Discourse Description (UoDD)
(John Doe)
Universe of Discourse (UoD)
John Doe
Reference

The speculative nature of the UoD/UoDD approach is plain from the description of its proposed implementation. It is clearly intended to be a description of what, ideally, an IS should be. It is not intended to be a description of what all information systems are given that there are few implemented systems with an embedded conceptual schema. The underlying framework is sufficiently clear and comprehensive, however, to allow an extrapolation from this proposed ideal to an underlying general theory of information systems: This theory suggests that they contain information that directly reflects the UoD, which has been transformed to accommodate technological concerns without affecting its basic semantics.

In turn, this suggests how information systems should be developed. The first step is to produce a specification of the UoD (business model) and then to transform this into a specification of the UoDD (system model). Given the referential assumption, this transformation is regarded as a straightforward re-interpretation of the specification as referring to the UoDD – one which requires no real change in content. A description of the semantics underlying the transformation needs to explain how the reference of the signs in the (single) specification appears to shift however. One could view the transformation as an interpreter deciding to re-interpret the signs in the (single) specification in a different way. Alternatively, one could regard the signs in the specification as having two references – one for the UoD and another for the UoDD and the transformation shifting between these. In either case, the key point that needs to be made is that there is a single specification (a single set of signs) that do two pieces of semantic work – being both a business (UoD) model and being a system (UoDD) model and this is achieved though a shift in the reference of the signs.

This way of thinking of the development of the Information specification is now generally accepted and can be found in standard textbooks. For example,[7] state that at the first stage a model is constructed by asking the customers what are "…the "things" that the application or business process addresses" (the business UoD model). Their work goes on to say "These "things" evolve into a list of input and output data objects as well as external entities that produce or consume data" (the system UoDD model). This description suggests a single specification where the analyst acts as an interpreter, who makes the shift in interpretation between the UoD and the UoDD. To complete the process, there is a subsequent stage where the technological concerns are taken into account and the system UoDD model is transformed into a physical model to be implemented. This second transformation provides a useful contrast to the first as it is a syntactic transformation - unlike the first transformation it involves no change in the underlying UoDD semantics, but usually a significant change in the syntactic shape.

Figure 2 – Different ontological and epistemic structures



Patterns of Epistemic Divergence

In our reverse engineering of working systems we have found a number of common patterns that challenge the referential assumption – patterns of epistemic divergence. For reference, the examples presented have been greatly simplified to clearly (a) raise doubts about the coherence of the UoD/UoDD's simplification (referential assumption) and (b) develop an understanding of the notion of epistemic divergence and how it is explained by the notions of epistemology and ontology.

Split Super-Sub-Type Hierarchy. The super-sub-type hierarchies are the backbone of a business ontology. While one would be forgiven for assuming that such a key structure is clearly reflected in IS, the reverse engineering of business semantics from working business information systems shows that their databases often contain deliberately split hierarchies reflecting a divergence between the epistemology and the ontology. For example, consider a business system that keeps records of the Users of the system for security reasons, and of Clients and Suppliers for the usual business reasons. Like many other systems, it keeps the records of its Users separate from its records of Suppliers and Clients. The system also explicitly recognizes that Suppliers and Clients are sub-types of the more general type Persons. The system records that John Doe is a Client and a Supplier: It also records separately that John Doe is a User, hence the system has no way of knowing that this John Doe is the same person as the Client/Supplier John Doe. From the information systems epistemological point of view, there are two representations of John Does. From an ontological perspective there is only one John Doe. This is illustrated in Figure 2, which clearly reveals the design decision to split the super-sub-type hierarchy and introduce epistemic divergence: It also clearly shows that the information (epistemology) in the information system is not a straightforward reflection of the business domain (ontology).

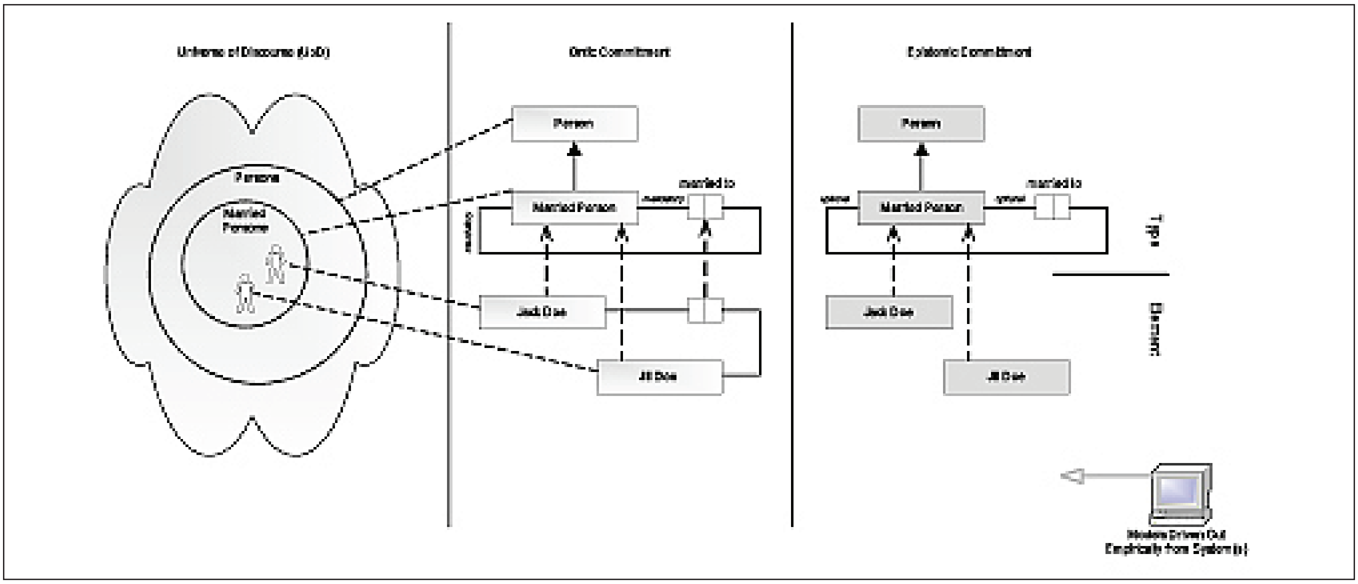
The ontology influences the shape of the epistemology, but it is not the source of the epistemic divergence. No amount of analysis of the business domain (ontology) would, by itself, indicate whether and where the hierarchy should be split. Similarly, no amount of analysis of the split hierarchy epistemology on its own would reveal the united hierarchy nor would analysis of the two representations of John Doe, by themselves, reveal they are of the same person.

Epistemic Transformation of Cardinalities. Relations are a vital component of business domains. It is commonplace when characterizing relations to indicate their cardinality – for example, whether they are mandatory or optional. This gives rise to another common pattern of epistemic divergence – the epistemic cardinality of relations. As an example consider an enterprise that has an information system that records the marital status – single or married – of its customers when they make a purchase. For one product, the system also needs to record details of the married customers' spouses. For customers of other products, the details of the spouse are not recorded (and it is a requirement that they should not be). So the epistemology has the notion of 'Person', its subtype 'Married Person' and an optional relationship 'married to' between married persons. The system keeps a record of Mr. and Mrs. Doe, who are married, who have independently purchased products. On their application forms they stated that they are married persons, but did not, and were not required to, provide details of their spouses. So the system has no record of their marriage relationship.

The ontological picture of this relationship is different. It is part of the definition of married that if someone is a 'married person' then s/he always has a married to relationship to their spouse, who is also a 'married' person. The ontic cardinality of the married to relationship is therefore mandatory not optional. The ontology and epistemology are pictured in Figure 3.

The structure of the epistemic divergence is clear. The cardinalities of the 'married to' relations are different and

Figure 3. Cardinality's epistemic divergence



a concrete illustration of this is that the ontology has Mr. and Mrs. Doe's married to relation and the epistemology does not. Again the information (epistemology) is not a direct reflection of the business domain (ontology).

It is important to be clear that the epistemology and ontology contain the same 'married to' relationship. We can develop arguments that the epistemology's 'married to' relationship is not the same as the ontology's – but these turn out to be unsatisfactory. For example, we could claim that the epistemological relationship is really a 'known married to' relationship – reducing the epistemic divergence. This explains why Mr and Mrs Doe's (unknown) married to relation is not an instance of it. This situation would have the odd counter-intuitive implication that the instance comes into existence when the enterprise learns of it, and not when the couple gets married: It also has the disadvantage of being dependent on the particular application ('indexed' in linguistic terms).

What the epistemic cardinalities describe is the particular system's epistemic content,[2] where 'optional' means it is optional whether the system has to know the ontologically mandatory relationship – not optional whether the relationship exists. In this case, Mr and Mrs Doe's married to relation is an instance of the ontology and epistemology's common married to relation, but not a known instance of it relative to the particular system.

Importance of Roles for both Ontology and Epistemology

Both these examples of epistemic divergence show that the referential assumption is a simplification; the assumption 'works' in some cases, where the epistemology is a good reflection of the ontology, but not in (many) others. The blanket referential assumption ignores the latter – and so the role epistemology plays in information systems development. We contend that, from the perspective of understanding a working system, both the ontology and epistemology have important roles to play. Ontology reveals the meaning more clearly while the epistemology specifies what the system knows. Importantly, one cannot deduce one perspective from the other. For example, one cannot work out that married persons always have a married to relation from the epistemology. This is the key reason why understanding is aided by having both an ontology and an epistemology and, of course, the relation between the two.

As things stand, information systems and their documentation typically do not contain a description of their ontology – most, at best, contain an epistemological 'gloss' on their ontology. From an interoperability/integration perspective this is problematic, as what different information systems 'know' of things in the business domain is often different and/or similar knowledge is expressed differently. Problems arise as a consequence. Once the ontological and epistemological concerns are recognized, however, we have found that business patterns that are more 'fruitful' from an interoperability perspective can be engineered. The outcomes of fruitfulness can translate to enhanced business in many ways. For example, addressing the epistemic transformation of cardinalities example in the context of an insurance system would allow product discounts to be automatically offered to Jill Doe if Jack Doe was a customer (or vice versa). This could not happen easily in the context of epistemic divergence.

Reflections on the Challenge

This article exposes epistemic divergence. In research terms, there is a need to clarify and deepen our understanding of the challenge and approaches to dealing with it. Though the examples above are enough to establish the existence of the phenomenon, we believe there is a consequent need to undertake systematic and detailed research to ascertain the:

1. *Types of epistemic divergence.* While the work here demonstrates split hierarchies and the transformation of cardinalities, our own work indicates that other types of divergence exist, which need to be classified, their nature analyzed and their uses (and implications) better understood.

2. *Prevalence of those types of within existing systems.* As an adjunct to the above point, the prevalence (and effects) of different types of epistemic divergence need to be systematically assessed in the context of working information systems.

3. *Metaphysical framework (ontological theory).* The clear division of epistemology and ontology starts from a clear (philosophical) notion of ontology. There is limited work in the community that examines the underlying metaphysical framework and the effect of metaphysical choice on the resulting ontology (in an information systems context). The need for this kind of research is recognized[9] and is important in overcoming noted limitations of ontology as it is currently conceived.[6]

More generally, there are several obvious areas for further research which are outlined below:

4. S*ystems development lifecycles.* Simply put, there is no explicit acknowledgement of ontological and epistemological concerns in current methodological lifecycles. Even a cursory examination of development methodologies shows that this choice has been overlooked, which raises the question of how we should incorporate it into the lifecycle development process.

5. *Methods and techniques.* In practice, technology heavily influences the methods and techniques used to develop information systems – typically these are organized within the framework of the implementation platform rather than any ontological or epistemological concerns. In contrast, both the business domain (ontology) and epistemology are completely independent of these technology concerns. While this was appreciated by the UoD/UoDD paradigm formalizers, they did not provide a framework to resolve this issue, and this is still not effectively addressed in practice. Consequently, there is a need to establish and refine methods and techniques for working at these two levels.

The conceptual analysis, modeling and design associated with addressing these concerns are not trite and require a given level of skill (which likely needs to be developed). Some authors note that such aspects are typically not handled well as things stand, and cite literature which confirms that standard conceptual modeling techniques often fall into disuse within organizations.[10] Further, some authors propose that such techniques provide a poor return on investment in modern emergent organizations.[8] These points accepted, the pertinent question is whether changing how things are will allow us to see these issues in a different light (that is, we ascertain that have been answering the wrong questions to-date).

Conclusion

This article challenges the fundamental assumptions of the way that we currently understand the relation between an application system and its business domain. Our analysis of these assumptions has been framed in terms of separation of concerns that underlies most mainstream development – and reflects our understanding of the nature of information systems. The analysis itself has sought to demonstrate that the relationship between application system and business domain is based upon a simplification. Epistemic divergence is the salient outcome of this simplification, which has been demonstrated via the use of simplified examples taken from commercial legacy systems. Unraveling this simplification provides a clearer and more precise understanding of what working Information Systems are – in particular, it reveals the finer grained set of concerns of ontology and epistemology.

The conclusion drawn from the exposition of epistemic divergence is that we need to fundamentally change the way we understand the relation between an application system and its business domain to provide a conceptual framework that can effectively explain divergence. From a research perspective, this conclusion provides us with a start point for clarifying and resolving the challenge that epistemic divergence holds. Accordingly, we have sought to provide an agenda to clarify this new framework and act as a basis for developing approaches that are significantly more effective in managing semantic complexity and interoperability. We believe epistemic divergence be important as we contend that the prevailing development paradigm does not have the conceptual technology (awareness and techniques) to aid semantic understanding as effectively as is necessary.

References

1. ANSI/SPARC, The ANSI/SPARC DBMS Model. In *Proceedings of the 2nd SHARE Working Conference on Data Base Management Systems, in SHARE Working Conference on Data Base Management Systems*, (Montreal, Canada, Apr. 26-30 1976), North-Holland Publishing Company, Amsterdam.
2. Chalmers, D.J. *The Conscious Mind: In Search of a Fundamental Theory*. Oxford University Press, NY, 1996.
3. Griethuysen, J.v. ISO/TC97/SC5/WG3-N695 - Concepts and Terminology for the Conceptual Schema and the Information Base. ANSI, NY, 1982.
4. Hirschheim, R.A., Klein, H.K. and Lyytinen, K. *Information Systems Development and Data Modeling: Conceptual and Philosophical Foundations*. Cambridge University Press, Cambridge, 1995.
5. Kent, W. Data and Reality: *Basic Assumptions in Data Processing Reconsidered*. North-Holland Publishing Company, Amsterdam, 1978.
6. Ouskel, A.M. and Sheth, A. Semantic Interoperability in Global Information Systems: A Brief Introduction to the Research Area and the Special Section. *ACM SIGMOD 28*, 1, 5-12.
7. Pressman, R.S. and Ince, D. S*oftware Engineering: A Practitioner's Approach*. McGraw-Hill, London, 2000.
8. Truex, D., Baskerville, R. and Klein, H.K. Growing Systems in Emergent Organisations. *Comm.ACM 42*, 8, 117-123.
9. Wand, Y. and Weber, R. Reflection: Ontology in Information Systems. *Journal of Database Management 15*, 2. III-VI.
10. Wand, Y. and Weber, R. Research commentary: Information systems and conceptual modeling - A research agenda. *Information Systems Research 13*, 4, 363-376.

Mark Lycett Mark.Lycett@brunel.ac.uk) is a Reader in adaptive information systems at Brunel University.

Chris Partridge (partridgec@BOROGroup.co.uk) is a Visiting Fellow at Brunel University.