7. Information systems technology grounded on institutional facts

Robert M. Colomb, ***School of Information Technology and Electrical Engineering, The University of Queensland***

Abstract

This paper presents a theory explaining the success of information systems development based on SQL-type database technology by showing that the assumptions underlying that technology correspond very closely to the way Searle's institutional facts are created. The theory presented is a theory of action and design, so its productivity is shown by retrodiction of the necessity for business process engineering to achieve integration of information systems within an organisation, and prediction that interorganisational integration of information systems using the internet can succeed only if the applications share institutional facts. The theory is used to predict that autonomous intelligent agent applications can succeed in the information spaces populated by these common institutional facts.

Introduction

Information systems are generally and very successfully implemented using a particular sort of technology typified by relational database systems, which I will call *logical databases* for reasons that will be explained below. There are alternative technologies. Why have logical database systems been successful?

Information systems have, for the most part, been successful in relatively restricted organisational subunits. A large organisation therefore may have hundreds of information systems. Over the past two decades organisations have been trying to develop information systems implemented by logical databases at the scale of the whole, typically by integrating the successful local systems. There are successes, but it has turned out that it requires an enormous effort, including changes in the way the organisation sees itself (e.g. through business process re-engineering), in order to achieve success. The question is: why is it so hard to extend successful local information systems to an organisation-scale system?

Organisations interoperate with other organisations in a global economy. A global communication infrastructure now exists which makes it easy for anyone to communicate with anyone else. There is a strong business case to interconnect the logical database-implemented systems of multiple organisations for a wide variety of purposes. But if it is hard to integrate systems within a single organisation, what hope is there for integration across organisations? After all, many of the things done to achieve single-organisation integration depend strongly on central management commitment. There is by definition no central management where the problem is to integrate systems across organisations. What can we hope to achieve?

We have a technology that works extremely well on a small scale, is difficult but possible to adapt to an organisational scale, and which we now want to further adapt to a global scale. The thesis of this paper is that in order to understand what is feasible on the

global scale we need to understand why the technology is so successful on a local scale, and why it is difficult to adapt to an organisation-wide or larger scale.

Success at the local scale

Why are logical databases the technology of choice for implementing information systems?

Information systems are generally about the management of records. Records can be records of just about anything: a company's accounts, medical records, criminal records, a census, the archives of a newspaper, or the contents of a museum. Just about anything can be a record; the Babylonians used clay tablets to record their business dealings, some medical records are images, most of the contents of most museums are physical artefacts of various kinds. But contemporary information systems are generally concerned with documents that contain most of their information in the form of text. Physical objects like the contents of a museum are generally represented in information systems by documents called catalogue entries.

So, more specifically, information systems are about the management of records that are documents containing information mostly in textual form. The general technology for processing collections of text records is the text database.

The model of information-seeking behaviour supported by text databases has the following steps:

1. The user has an information need.
2. The user formulates the information need as a query consisting of a collection of terms.
3. The system returns the subset of its collection of documents containing all and only those documents that contain the query terms.
4. The user then reviews the documents returned, and makes a judgment as to whether each document satisfies the information need or not. The expectation is that many of the documents returned will be irrelevant (*limited precision*). The expectation is also that some of the documents in the collection that would have satisfied the information need were not returned, because the query did not contain appropriate terms (*limited recall*).

Precision and recall are measured on a percentage scale. A precision of 0% means that none of the documents retrieved met the information need. A precision of 100% means that all did. A recall of 0% means that none of the relevant documents were retrieved. A recall of 100% means that all were. Returning the entire collection guarantees 100% recall, but gives a very low precision. Text database systems are considered to perform very well if their average precision and average recall are as high as 40%.

Computer-based information systems generally make use of technologies such as relational databases. There is a wide variety of such systems, but they are generally characterised by data models based on classes and instances, with relationships among classes. Typically the data model is expressed in a language like UML, one of the varieties of entity-relationship modeling, or object-role modeling. The populations of particular systems are generally managed by systems based more or less on the first-order predicate calculus, such as relational database systems or object-oriented database systems, which we here call *logical databases*.

In text database terms, a query on a logical database is expected to have 100% precision and 100% recall. A class list is the definitive statement of which students are enrolled

in a course. A person may attend lectures, submit assignments and sit an examination, but if they are not on the class list then they are not enrolled and cannot be assigned a grade. Another person may never attend classes, submit no assignments and not sit the examination but, being on the class list, is considered enrolled and will be given a grade, perhaps one signifying 'no assessment submitted'.

Because a query on a logical database returns all and only the documents satisfying the information need, it is possible to construct much more complex queries. Combining information from two different tables requires 100% precision and 100% recall. So does the reliable use of negation, and complex selection conditions.

The claim here is that logical databases are the preferred technology for managing collections of records using information systems. But all we have established so far is that an information system manages a collection of records. We need to look at these collections in more detail.

Consider a particular kind of collection of documents that are records of activity of an organisation, namely the correspondence incoming and outgoing. Imagine we have a UML model for this collection, and consider a particular document, namely a letter from a potential customer enquiring about the possible existence of a product that the company does not at present supply. Call this letter Q. We want to compare this with a letter from an established customer placing an order for an existing product. Call this letter P.

We want to look at what the organisation can do with letter Q compared to what it can do with letter P. Letter P can be cross-referenced with other documents associated with the established customer, and with other documents associated with the existing product. Some of the former will be invoices, statements, payments, and so on. Some of the latter will be picking lists, shipping orders, purchase orders and so on. The organisation will have standard queries associated with these documents, for example all orders that have been delivered but not paid for, or all orders for a customer that have not yet been shipped.

By contrast, it is not at all clear what to do with letter Q. It might routinely be answered with a polite negative reply. If the prospective customer will potentially place large orders, the letter might be sent to the product development group for a feasibility study. The product may or may not be technically feasible. If technically feasible, there may or may not be the capital available for development, or there may be higher return uses for the capital that could be used for the project. It would be hard to know with what other kinds of documents letter Q would be associated, and hard to see what routine queries might retrieve it.

Letter P fits well into the class/instance/relationship data model, while letter Q does not. The class/instance/relationship data model permits the construction of complex queries, the reliable definition of negation, and so on. Information systems generally exclude documents like letter Q from consideration, concentrating on documents like letter P.

So, the preliminary answer to the question as to why information systems are implemented using logical rather than text databases is that the subset of records considered by information systems are very largely those that are usefully modeled using the assumptions underlying logical databases, and so can profit from the much richer querying capability of logical databases.

However, this is hardly a satisfactory explanation since it is circular. Information systems use logical databases because they are about managing the sorts of records that can be

well managed by logical databases. We need a deeper understanding of these sorts of records.

If logical databases are the solution, what is the problem?

What characterises logical databases in relation to text databases is that logical databases need the concept of logical equality and the subsumption of individual by class, so the data for which a logical database is to be used must support these concepts. Text databases do not make these assumptions. This is the reason text database systems suffer from problems of limited precision and limited recall.

For an object to be represented in a logical database, it must be completely characterised by the classes of which it is an instance. Letter P of the previous section is completely characterised by its membership in the class *order* and its membership in associations between the class *order* and the classes *product*, *customer* and so on. To the university student record system, a person is completely characterised by membership in the class *student* and membership in associations between *student* and the classes *enrolment*, *program* and so on. This is why we can expect 100% precision and 100% recall.

In a text database, we can't even reliably identify a document as a member of a class, much less characterise its content by class and association.

The ability to completely characterise an object by the class in which it is an instance is the basis for logical equality, which in turn is necessary for the computations performed in logical databases. The number of students enrolled in a course can be computed because the class list defines the enrolment, and all students' enrolments are equivalent. A grade point average can be computed because a student's performance in a course is completely characterised by the grade awarded, and the same grade awarded in different courses is logically the same.

So the first answer to the question as to what problems a logical database is a solution for is those applications where the assumptions hold that class and association membership completely characterise the objects. This might be somewhat less circular, but is still not satisfactory. What sort of world produces records that are completely characterised by class and association membership?

What sorts of applications satisfy the requirements for logical databases?

The world is a messy place. We tend to make order in it by classifying things. Most animals classify the world into at least the categories *food*, *predator* and *mate*. But these sorts of classifications are not enough for logical information systems since they do not completely characterise the objects in the world. A botanist may classify a forest by genus and species, but there is room for error. Observations of specimens in different ways can lead to a change in its classification. The object in the world is primary. We can use logical databases for applications like this, but we have to ignore the individual objects and treat them only as instances of classes.

We need to keep in mind that our information systems contain not the world, but statements about the world. That is, Popper's third world (McDonald, 2002). (Popper's first world is reality, his second is internal psychological states caused by an organism interacting with the first world. The third world is what the organism says about its experience.) Both letters P and Q are in the third, as well as the first, world.

What differentiates letter P from letter Q is that letter P is an instance of an institutional fact as described by Searle (1995). An institutional fact is a statement about the world, but the world it is a statement about is a social world. It has no meaning apart from the

society in which it occurs. (There are enormous differences in approach between Popper and Searle, but at a first approximation, the claim that an institutional fact is one kind of statement about the world seems reasonable.)

Searle distinguishes institutional facts from brute facts. A brute fact is a statement about something in the world outside of human society. Examples of brute facts are: 'Thylacines are extinct', 'Canberra is cold in the winter', 'This is a 2.5 centimeter diameter gold-coloured metal disk', 'This is a piece of white paper with black marks on it'. All of these statements would continue to be true if our society disappeared. (Of course there would have to be some sentient being to make the statements, perhaps robots or extraterrestrials.)

All objects, including statements, are for Searle brute facts. A written statement can be black marks on white paper. A spoken statement is acoustic waves in the atmosphere at a particular place at a particular time. What makes a brute fact an institutional fact is how it is taken by the people concerned about it. In particular, an institutional fact is taken as a record of an instance of a standardised speech act performed by a social institution in a human society. A 2.5 centimeter diameter gold-coloured metal disk is taken to be a dollar coin in Australian society in 2004. A piece of white paper with black marks on it is taken as an order for particular goods by Acme Manufacturing Company at a particular time.

Searle's formulation starts with speech acts. A speech act is an action made by a designated person on behalf of a social institution that changes the social reality managed by that institution. The quintessential speech act is giving a new baby a name. The action is entering writing in blank spaces on a form, then lodging the form at the office of the Registrar of Births in the jurisdiction in which the baby was born. The designated person is one of the parents of the baby. The form is supplied by the Registrar of Births. The form is lodged by handing it to a designated officer of the Registrar in their designated office during the designated office hours. The social reality changed is that a new person now exists with the name indicated on the form. The institutional reality managed by the Registrar of Births is the population of citizens of the country of whose government it is an arm. That the person into whom that baby develops is named its name is an institutional fact. Records of this institutional fact are stored by the agency and on birth certificate and passport documents, but also exist in people's memories and are created whenever the name is used, especially in other official documents.

Searle's formulation is 'brute fact X counts as institutional fact Y in context C'. In our naming example, the brute fact is the filling in and lodging of the form. The institutional fact is that the baby has the designated name. The context is everything else: the person lodging the form is a parent, the office is the proper office, the form is given to the proper person at the proper time, and so on.

What most clearly differentiates letter P from letter Q is that letter P is an institutional fact. Sending and receipt of letter P by the appropriate people counts as the speech act of placing an order. When this occurs, the world changes, in that the receiver of letter P (the supplier) is entitled to ship the nominated quantity of the nominated product to the sender (the purchaser) and expect payment in return. The copy of letter P (brute fact) held by the supplier is a record of the institutional fact of the purchase order having been made. The context includes the supplier being in the business of selling the nominated product, the purchaser being a properly constituted customer, and so on.

The whole business is regulated by the laws of commerce in the relevant jurisdictions. In addition, it is regulated by a body of largely implicit customary practice. This body

of customary practice is called *background* by Searle. Background is a significant aspect of any context.

Institutional facts are a subclass of what Searle calls *social facts*. Social facts are informal, while institutional facts are formal acts of formally constituted institutions. That my nickname is 'Bob' is a social fact, but that my official name is 'Robert Michael Colomb' is also an institutional fact. 'A is a friend of B' is a social fact, but 'A is the spouse of B' is an institutional fact as well. 'A is influential' is a social fact, but 'A is prime minister' is also an institutional fact. The institution or network of institutions that provides the context for institutional facts is a complex system of social behaviour. Different institutional environments have different informal patterns and norms of behaviour (*culture*) that are the background aspect of the context of the institutional facts it creates and maintains.

One key characteristic of institutional facts, at least in our present society, is that they are designed to be completely characterised by the classes to which they belong. Every name is completely characterised by the speech act of registration with a birth certificate as record of the institutional fact of having been named. Every purchase is completely characterised by the various classes by which the supplier and purchaser do business. Every student is completely characterised by the program and courses in which they have enrolled. This is the defining feature of modern bureaucracy. This is the reason people worry about 'being just a number'.

Nearly all information systems are used to store, retrieve, and now often create institutional facts. Society agrees that nothing is relevant except that 'brute fact X counts as institutional fact Y in context C'. There are a finite number of well-defined context types. All contexts of the same type are the same, so all institutional facts resulting from these contexts are the same. To make this work requires a highly disciplined form of behaviour, and a rigorous enforcement of the framing rules defining the contexts. This is the reason for the complex system of commercial law, standardisation of accounting rules, requirements for audit, and so on. But the standardisation also relies on the informal behaviour patterns and norms constituting the background.

That institutional facts are completely characterised by the classes constituting the operating rules for the institutions creating them corresponds exactly with the assumption underlying logical databases, that their contents are completely characterised by the classes of which they are instances. I submit that this is the reason for the overwhelming dominance of logical database technology in information systems.

In the following we are going to need some perhaps unfamiliar terminology. An *ontology* is a representation of the world with which a system is concerned. The rules of chess or cricket are an ontology. For an information system, the ontology consists of its data model, business rules, and a characterisation of the individuals with which the system deals. An ontology is *transcendent* if it contains the constituting rules for the relevant behavioural interactions, and the routine behavioural interactions cannot change their constituting rules. The rules for chess or cricket or the grammars of programming languages are transcendent. An ontology is *immanent* if the routine behavioural interactions can change the rules. Human natural language is an immanent ontology, since the grammar rules are patterns abstracted from practice and practice can change them, albeit slowly. The ontology of news topics in a newsfeed change as events happen in the world. The ontology given by the directory structure of a person's personal computer is immanent, because the user of the computer is free to change the directory structure.

The schemas defining types of institutional facts define a transcendent ontology for the information systems supporting the creation of institutional facts and keeping records of them. Data models for particular systems are representations of and implementations of aspects of the ontology. The technology implementing these data models works only because of the behavioural disciplines that implement the framing rules of the various speech acts. If each letter placing an order requires separate consideration and is treated in a unique way, the order entry system of the supplier can't work the way we expect it to. But of course the transcendent ontology is only the formal part of the system. The context of all speech acts includes the background, which is characteristic of particular institutions and differs between institutions.

How does this view help?

The theory described in this paper can be considered as a theory for design and action in the taxonomy of Gregor (2002). As such, it should be useful in guiding future designs.

One thing the theory does is explain why SQL and other logical databases are overwhelmingly the platform of choice in information systems implementations. This, however, does not seem to be a controversial situation. It is not a matter for concern, and there are no serious proposals for any other kind of platform. So to have value, the theory in this paper must do more.

The success of logical databases in information systems is most apparent in systems that serve highly focused organisational subunits. These are the subunits responsible for limited classes of speech acts, so needing records of limited classes of institutional facts. These are also the levels of institutional structure where the informal behaviour patterns and norms are the most stable, so where the background aspect of the context for the institutional facts is the most uniform.

As a result of success at this scale, there has been for many years a push to tie the information systems together. More recently, the availablity of cheap and powerful communication facilities has led to a push for tying together information systems of separate organisations into what may be thought of as world-scale computing. Although there have been successes at both of these enterprises, there have been many failures, with projects abandoned after vast expenditure. The idea that logical databases work well because they manage institutional facts can explain the successes and failures, and can be used to predict *a priori* whether a given project proposal has a chance of success.

The first of these enterprises, that of tying together the information systems in a single large organisation, was given a formulation as an extension of logical database technology in the federated database movement whose strategies are summarised by Sheth and Larsen (1990). The idea was that if we had many individual information systems, we could build a single big system by federating the data models and schemas of the local systems without requiring changes in the local systems. These efforts often failed, an example being the CS90 project of Westpac Bank in Australia in the late 1980s, which was abandoned after several years at a cost reported to be about A$500 million. Other major banking projects of the type were similarly abandoned at even higher costs.

In terms of the present theory, the reason these projects failed is that the speech acts performed by the organisation did not extend to the appropriate scale. The organisational subunits are in fact generally created to perform the limited class of speech act, and the framing rules for the speech act are often largely limited to things within the scope of that organisational unit. In a bank of the 1970s the savings accounts would be managed by a department, which would define what a customer was, the rules for interest pay-

ments, what addresses were kept, and so on. The home mortgage department would have analogous definitions, but there would have been no mechanisms to synchronise them. Also, different types of speech act have framing rules that take different things into account. A two-year-old might be a valid customer for a savings account, but not for a home mortgage, for example. Large organisations typically support hundreds of separate information systems serving low-level organisational units or specialised staff functions, and the speech acts performed by these subunits are typically uncoordinated. Furthermore, each organisational subunit has its own culture, so contributes a different background to the context of the speech acts for which it is responsible.

To integrate the information systems supporting these organisational subunits required far too much negotiation and resolution of different views of what were in principle common concerns, beyond what was needed to support the speech acts for which these units were actually responsible.

Tying together the information systems of a large organisation turned out not to be primarily a technical problem. It did require a large investment in technology, but was also predicated on extending the scope of the speech acts performed by the organisation to encompass all of the interactions needed to serve particular stakeholders. This involves not only the formal rules but requires creating a common culture so as to create a uniform background. This extension of scope is called business process reorganisation. If a bank wants to provide a web interface integrating all the services it provides to a given customer, the various departments need to come to a common definition of what a customer is, how they are named, what addresses they can have, under what conditions a customer is enabled to access a particular product, and so on. Making these decisions then reorganising the organisational subunits to work from the now larger scale ontology is a major cost to the organisation. Investment in technology is an enabling factor for business process reorganisation, but is not the major cost.

The prediction of the theory is that no proposal to integrate the separate information systems of organisational subunits is likely to succeed unless the organisation is rebuilt so that the speech acts it performs are at the scale of the whole of business interaction with classes of stakeholders. Once the speech acts are at the right scale, the consequently revised schemas and models will be able to be integrated in a relatively straightforward way. So the failure of the federated database approach to information systems integration can be retrodicted by the theory.

A similar problem has arisen more recently with the Internet. Since it has become technically feasible to interconnect systems operated by different organisations, people have been talking about interoperation. Of course people have been able to find resources using text database technology (search engines), and to compose individually selected services for particular purposes, but the dream is to be able to interoperate automatically using logical database technology. (This is often called use of intelligent agents.)

There are a number of manifestations of this dream, the most recent and concrete of which is the semantic web (Berners-Lee and Fischetti, 1999). There are a fair number of developments of what might be thought of as infrastructure for interoperation, for example XML, RDF, OWL, SOAP and WSDL [1] . There is a sometimes not clearly expressed dream that if you represent your web site or database in XML, or if you put descriptors on your site using RDF or OWL, then you can interoperate using logical database technology with anybody else who does so too.

[1] More details of any of these can be obtained from www.w3c.org

The theory of this paper, that logical databases work because they store institutional facts, leads to the conclusion that interoperability using logical database technology is only possible if the interoperating sites share speech acts and consequently share in the creation of institutional facts. In particular, they must have sufficient shared culture so that the background is sufficiently uniform. (For another view of this issue, see Colomb, 1997.) Some of the kinds of situations where this condition is satisfied include:

1. The sites do business together. This is what Electronic Data Interchange (EDI) is all about. For example, a group of businesses agree on common terms and common business messages with agreed semantics, and can then buy and sell from each other by the interoperation of their respective purchasing and order entry systems. E-commerce exchanges are built on this basis. The agreement on common terms and common business messages with agreed semantics constitutes the synchronisation of the framing rules for speech acts, so that the interoperation can make speech acts and there is an agreed semantics for the consequent institutional facts. The agreement is a transcendent ontology, supported by a common background. The ontology is transcendent because the only way to change the common world is to change the ontology, which is done by the management body outside of the routine interoperation of the sites.
2. All sites report to a central body using a common ontology. Tax returns in a given jurisdiction or financial reports to a given stock exchange are examples. The common ontology is the set of regulations and accounting standards established by the tax office or stock exchange and enforced by auditors and the commercial law institutions. This ontology is generally transcendent because it is imposed by the central body, and the participation in the relationship with the central body gives aspects of common culture so there is a stable background.
3. All sites operate as small players in a dense market. An example is residential property sales in a particular city. There are many agents, many sellers and many buyers, and each has the choice to deal with many of the others. In these markets, conventions develop so that to do business one must do it pretty well the way everyone else does. The speech acts and consequent institutional facts are similar by convention rather than by agreement. Any innovation either dies out quickly or is quickly adopted by everyone else due to competitive pressure. Here, the ontology is not transcendent, but immanent, derived from patterns in the background. It is possible to build, for example, services that will search for a house in many agents' sites. There are many ways to do this requiring more or less cooperation among the players. An immanent ontology is unstable in that a player may innovate at will, and that innovation may take off unpredictably. Background is the critical factor in this situation.

Unless there is some reason to assume the interoperating sites share institutional facts, there is no reason to think that interoperability using logical database technology is possible.

How can we build on this?

Our theory leads us to expect that we can build interorganisational information systems using logical database technology, enabling interoperation among organisations that share institutional facts. The sharing of institutional facts is represented by the participants' commitment to a common ontology. This ontology can be either transcendent or immanent. The question now is: given that we can interoperate where can we then go?

One possibility is to recognise that once an interoperating community is established, it can generate a large number of institutional facts. These institutional facts can be interpreted by any of the players who share the common ontology. These ontologies or institutional fact schemas constitute the atomic behavioural units, but do not necessarily determine behaviour. The rules of chess determine what constitutes a chess game, but there are lots of different games.

So we can use techniques like data mining that depend for their atomic data on the exact classification/logical equality nature of institutional facts, but which can find emergent patterns in the multiplicity of instances. These emergent patterns can be used as an immanent ontology for strategic or tactical decision making, for example advertising campaigns to encourage or discourage behaviour patterns, or as evidence of undesirable behaviour to be subjected to further investigation (e.g. fraud, money laundering).

Where the interoperating community consists of many small players, there may be an advantage to each player giving up its exclusive access to the institutional facts it creates in favour of a community-wide pool to which all players have common access. This is common, for example, in real estate where individual sales reports and auction success rates can be published for a whole city market area, enabling each player to see trends to which they can respond in their own fashion.

The information spaces opened up in this way give great scope for the development of interoperating autonomous intelligent agents. Each agent can develop its own immanent ontology, which it uses to govern the strategies and tactics it uses to interoperate with others to perform speech acts using the common transcendent ontology. The theory of this paper predicts that a research and development program along these lines would be likely to be productive.