**The CIDOC Conceptual Reference Model**

Stephen Stead

## About

**The CIDOC Conceptual Reference Model (CRM)** provides definitions and a formal structure for describing the implicit and explicit concepts and relationships used in cultural heritage documentation.

The **CIDOC CRM**is intended to promote a shared understanding of cultural heritage information by providing a common and extensible semantic framework that any cultural heritage information can be mapped to. It is intended to be a common language for domain experts and implementers to formulate requirements for information systems and to serve as a guide for good practice of conceptual modelling. In this way, it can provide the "semantic glue" needed to mediate between different sources of cultural heritage information, such as that published by museums, libraries and archives.

The **CIDOC CRM**is the culmination of over 10 years work by the [CIDOC Documentation Standards Working Group](http://www.willpowerinfo.myby.co.uk/cidoc/wg1.htm#docwg)and[CIDOC CRM SIG](file:///Y:\tmp\georgis\cidoc_tutorial\who_we_are.html) which are working groups of [CIDOC](http://www.willpowerinfo.myby.co.uk/cidoc/). Since 9/12/2006 it is official standard [ISO 21127:2006](http://www.iso.org/iso/en/CatalogueDetailPage.CatalogueDetail?CSNUMBER=34424&scopelist=PROGRAMME).

## The author

**Stephen Stead** was a founder member of [CIDOC Conceptual Reference Model Special Interest Group (CRM-SIG)](http://en.wikipedia.org/wiki/CIDOC_Conceptual_Reference_Model). He has worked on the CRM since 2000 and is one of the editing team that guided the standard through the ISO standardization process. He has delivered tutorials on the CRM in more than a dozen countries, including Korea, Brazil, Greece, Russia, and the USA. He is a graduate in *Archaeological Science from the Institute of Archaeology, University of London*and has worked as an independent consultant in the Heritage sector since 1991.He is a visiting lecturer at the*University of Southampton* and the *University of Kent at Canterbury*. He has served on the *CIDOC* Board since 2001 and on the *International Steering Committee of the Computer Applications and Quantitative Methods in Archaeology Conference (CAA)* since 1989. He is also active in the FRBR harmonization process.

He has worked on Heritage sector data analysis and design for over 20 years and has a number of designs in common use. These include *the Integrated Archaeological DataBase (IADB)*, the *Record of Scheduled Monuments (RSM)*, the *exeGesIS HBSMR (Historic Buildings, Sites and Monuments Record)*, the *British and Irish Archaeological Bibliography (BIAB)*and the *CIDOC International Core Data Standard for Archaeological and Architectural Heritage*.

**E-mail:** [steads@paveprime.com](mailto:steads@paveprime.com)

## Slide 1

Welcome to the CIDOC CRM tutorial. We are going to go through how the standard was built up, the principles behind it and the contents of it and hopefully at the end of this you'll be able to use the CRM in your own research to take you forward, and combine the different data streams that you have available to you and bring them out as an integrated whole.

So, the idea behind the CRM is that it is a standard which will help you integrate cultural information from a variety of sources and bring them together into an integrated environment where you can ask bigger questions than you can ask from the individual pieces that you have.

My name is Stephen Stead, and I’m a member of the CIDOC Conceptual Reference Model Special Interest Group. CIDOC is the Comite Internationale pour la Documentation <the International Committee for Museum Documentation>, part of ICOM (the International Council of Museums). They have hosted the development work over the last 15 years of the CIDOC CRM, and acted as a place where we can gather people from multidisciplinary teams to give us a range of inputs into CRM. So the CRM is not just a Museum standard, it’s a standard that reaches across museums, libraries and archives and it's fully multidisciplinary in the way it’s been put together. We've had everything from physicists through to archaeologists, which is a fairly wide mix of different people. Disciplines from all over the world have contributed to this, so that it's not North American or European-centric, it's a truly international multidisciplinary approach to developing a standard. Over the next few hours I'll take you through the way the standard was conceived, the methodology we’ve used and the contents of it, so that hopefully you'll be able to see how to use it. I’ll give you a few examples of applications of it, for projects that we've seen and used already.

## Slide 2

To get us started, I'll set the scene by giving us a problem statement, which will talk about the diversity of information that we have in the cultural heritage domain and how we’re going to bring that together as part of the desire to tell bigger stories. The CRM is designed to enable us to do that - we’ll look at an example of how we might use it to give us some motivation, a way of looking at the different data streams and how we’re going to combine them to produce a richer environment for us to ask questions in. Then we’ll talk about what the CRM was intended for, what things it doesn’t do, and talk about the methodology behind how we’ve made it and the form that it takes - what decisions there are, what design processes there are in there, and what they resulted in. And then we’ll look at the contents of the standard, so we’ll actually go through very different pieces of the standard, how they are work together to give us this integrating environment. Then we’ll talk about using it – we’ll give you some examples of its application in different fields. Finally, I’ll finally talk about the state of development: where we’ve got to, what the future holds for us and then a few conclusions to round the session off.

## Slide 3

So here’s the problem – we have cultural information from a very large domain and we have different types of information from different subdomains, different communities within the overarching cultural heritage domain, and they have very different sets of data that they want to integrate together. We have things like collection descriptions from a variety of institutions; we have collection descriptions of fine art; of archaeological-cultural heritage; physical cultural heritage information; we have information about natural history and geology; industrial processes - a wide range of information about a wide range of objects that we hold in our collections, the things that we’re interested in. In addition we have information in archives and literature, where we have the records about things that have been done, so we have copies of treaties, of contracts; we have letters written between people; we have artful works - things like poems and novels and plays - which have been produced as part of our cultural heritage, and we have records about the them as well. Now that’s given us things about the stuff that we have in our collections. We also need to administer those collections and preserve the objects and conserve them, so we have a wide range of documentation about the process of looking after those things and making them available to the public, about how we manage their preservation, and the actual processes of their conservation and preservation. We’ll have records about what treatments have been made on particular papers, what treatments are being made on objects, and how those have changed over time. So that gives us a record of the things that we have and how we’ve looked after them. And then we start to interpret them, to add to our knowledge about them, and this can be generally thought of as investigation and interpretation - the science and scholarship area. And we produce lots of new knowledge from the data that we hold about the things. So we’ve got data about them and how we’ve managed them, and then we start looking at them and developing knowledge, telling stories about the past and about human behaviour in the past, which we then wish to communicate. The final piece in the jigsaw is the presentation of this material, and that presentation might be to a group of our peers who have similar interests, and we are trying to convey the detailed knowledge that we've produced in our scholarly research, or it might be to a wider public, where we're trying to educate in some way. And that education might be to children or to adults as part of their continuing investigation of the world around them - hopefully that's something that carries on throughout their lives. And we’ll do that either as publications, which might be a scholarly treatise for our peers, or it might be a coffee table book with lots of colour pictures for a more general public. We’ll have teaching materials that we’ll produce for different communities, from adult education through to kindergarten, and we’ll have exhibitions which we’ll put on in the museums and institutions that we produce catalogues for and guides to and so on. So we have a wide range of presentation materials which we also want to integrate into our environment.

So we’ve got all of these different strands of information, enriched by the fact that there are many different subdisciplines, subdomains, working in different parts of the cultural heritage sector. They’ll have their own particular areas of interest, their own particular research methodologies, their own particular types of research question they want to ask. That adds another rich web to this rich set of datatypes.

So how do we make a documentation standard that allows us to bring all of this together? Well, it’s kind of difficult, because each of these different aspects has different methods, different forms, different communication requirements to disseminate and gather in information, in order to make the story rich. And while we find that there's lots of overlap in the data we’re interested in, we find it very difficult to build one schema which will hold all of it. We’ve tried - in the early 90s a very large project produced an entity relationship diagram, a relational model for museum data, and it was huge: it covered an entire wall with the table structure, and by the time you'd worked from the left-hand side to the right-hand side, you'd forgotten what there was on the left-hand side. Many discussions during the development were repeating discussions that had taken place a couple of years before, as we developed something further to the left. So it was a very complex environment and we found that this entity relationship diagram and this relational model were so big that very few people could understand it, and when we try to deploy it, we found that effectively nobody could use it. It *wasn’t* rich enough to actually hold all the information that we wanted to hold, but it was too complex for anybody to actually understand. So in a way it failed, but in another way succeeded, in that it showed us that we had to develop in a new way - and that new way was what we've used to produce the CRM. So we went for an object-oriented approach rather than a relational approach, and we started to look at a new way of bringing together information and understanding how it related to each other, how different pieces related to each other. We found out from work that was done at the Benaki Museum in Athens that a really powerful way to understand things that have happened in the past is to understand the relationships between those things and where they’ve met in time and space, so we can think of all of the things that have happened to our objects - our administration of them, our scholarly research, the things that happened to them in the past - as being <part of> events which have had different things there, and which have therefore expressed part of a story. So we can understand lives from the relationships between things and where they've met. And this began to give us a totally new insight into how to develop a standard which would give us access to this rich web of information that we had, but which we were finding very difficult to organise in any traditional way.

## Slide 4

Let's have a look at an example that will take us through the problem in more detail, and look and show what CRM does to help solve that problem. Here’s a very well-known document – it’s the protocol of the proceedings of the Crimea conference. This was a conference held between the Allied powers in 1945 to decide how they were going to divide up Europe and Japan after the Second World War. They could see that they were going to win, and they wanted to make a decision about which spheres of influence different parts of the Axis powers would be placed in. The meeting was between the Premier of the Union of Soviet Socialist Republics, the Prime Minister of the United Kingdom and the President of the United States of America, and they drew up this document, which declares the way the world will be after the end of hostilities. And if we look here we have a Dublin Core record for this famous document. It’s a text, it’s called the Protocol of the Proceedings of the Crimea Conference, subtitled the Declaration of Liberated Europe, and it’s dated 11th February 1945. It was created by these three great men, or the posts they were in. It was published by the State Department in the US and its subject is the post-war division of Europe and Japan. And that points to this document - it’s the Dublin Core record for this document.

## Slide 5

Now there is a famous photograph of the three great men at this conference, and much has been read into their body posture and the way they do or don't look at each other, and where they're staring at. We’ve got Stalin, Roosevelt and Churchill here, and this photograph you can actually get from Microsoft Encarta. The Dublin Core record for it is: it’s an image, it’s the Allied leaders at Yalta, it’s dated 1945, published by United Press International, and they never attributed anything to their individual photographers, it was always a UPI photograph, so there’s no individual photographer. It’s in the Bettmann Archive. The copyright for the Bettmann archive is given to [??] Corbis, and Corbis have licensed it to Microsoft for the Encarta, and here it references Churchill, Roosevelt and Stalin, different references to the same people - now we talking about their surnames rather than the posts that they held at the time they generated the document. Now, how do we get this photograph of the meeting and the document that they produced at that meeting together - how do we join them together? Well, if we look at the Dublin Core records, there’s not very much which allows us to get to the fact that this image is about the event where the document was created. If we look backwards and forwards between these two slides, the only thing that they have in common is the year 1945 – that’s quite a big search term. If you put 1945 into any search engine, you’re going to get a huge number of hits; it’s going to take a very long time to sort through them, and the chances of you actually noticing that this photograph and this document are related is very slim. It’s a big task, it’s going to take you all your life, and there are more important things in life than looking for a photograph of the event where somebody wrote a document that you're interested in. So probably not very much is going to happen there - but what happens if we add another data stream to these two streams, and try and get them to link together?

## Slide 6

Well, let's take another data stream. Here we've got the TGN - the Thesaurus of Geographical Names, produced by the Getty - and it goes through and talks about places. It gives us an identifier for them and then it says that there are different names for this place, with different spellings; it says that it's an inhabited place or a city; it gives us a latitude and longitude; and then it places it in a hierarchy - so it's in Europe, it’s in the Ukraine and it's in the Krim- the Crimea. And it’s the site of the conference between Allied powers in World War II in 1945. So yet another different way of expressing this meeting – it’s in the TGN, the Thesaurus of Geographic Names. And here’s a picture of Yalta. Now, if we go and look at *another* record in here - the record for Krim, we’ll find the alternative spelling of Crimea, so now we have a way of joining together the place – Yalta - with Crimea, where this conference took place in Europe. So this gives us a way of joining things together by using yet another data stream. So instead of more datastreams making life even more complicated, more data streams allow us to produce a link that allows us to get from one object to another object through intermediary steps, intermediary data sources. And that helps us produce an interesting picture of the world, a picture of the world which allows us to ask bigger questions.

## Slide 7

But there are a couple of problems that we need to deal with. The first is the identity problem – it’s part of our integration. We have different actors, and they have different names and different roles, all of which come together to produce a problem as to which person is which. How do we know that a document that’s talking about Joseph Stalin is also the same thing as an allied leader - how do we know that Stalin is the Premier of the Union of Soviet Socialist Republics? We need some way of bringing together these different identities and producing a single point, which tells us where that person is or who that person is. We’ve got a similar thing with different spellings of place names – Jalta and Yalta, Krim and Crimea - and we need to bring together different events: how do we know that the Crimea Conference is a different way of saying that the Allied leaders were at Yalta, which is a conference between Allied powers about the post-war division? These are all events which we need to identify as being one thing, so that we can relate different things to them. And of course we have different types of objects and documents that we need to be able to point to consistently across our environment, our web of information. We’ve got the photograph and the agreement text – so that’s one level of problem that we have to deal with.

## Slide 8

Another level of problem is that we have things which are hidden, which are not in the original records, which are hidden behind things. We don't know that Yalta and Crimea are places, that there is a relationship between Yalta and Crimea from the original documentation. We have to rely on the third set of documentation to give us the information that Yalta is *in* the Crimea. How do we know who the Allied leader is, or who the Allied power is? These are hidden people: we’ve got the title for them, but we don't know who it is that’s behind that title, and there are many variations of the way that title might be written, or that hidden name might be disguised. And again we have a similar problem with events. So how do we find the real thing and how do we identify it? The two of these <are> big problems. So what we need are the elements that are going to enable us to make those problems go away. How do we solve them? Well, we need the correct meta-data elements and <we need to> understand the relationships between those meta data elements in order to bring together the different data streams and get them to talk about the same thing. Or rather, they are talking about the same thing, but <we need to make them> enable us to identify the same thing in these different records and bring them together so that we can produce bigger stories.

## Slide 9

So let’s have a look at an example of how we might do that for the Yalta problem. Here we have some events which are going to help us bring together these different strands of information. We have an activity, which was the conference itself, and during that activity there was another activity, which was the creation of the document - the proceedings of the conference. So this creation event created the Yalta Agreement. The activity took place at some time within February 1945, and the creation was ongoing throughout 11th February, the date of the Yalta agreement in 1945. And the different actors and the different people – here I’ve just used a thumbnail of the person’s face to represent them – the three of them participated in the Crimea conference and performed the creation of the Yalta agreement. This Crimea conference is referred to by this image, so this image is documenting the conference, and the activity took place at place 701 2124, the TGM reference for Yalta. So that gives me access to all of the hierarchy of the Yalta records and Crimea records in the TGM. By using these events which we've created within our data structures, we are able to link together all of the different aspects of those two or three different data streams and produce a linkage which allows us to understand the relationships between the actors, the time and the place, and the things that have been produced: the documents and photographs. And by doing that we can now start to hook other information into here as well: maybe we know the names of other people who were at the conference and we can bring those in. This central activity allows us to link together a whole range of different information sources which are about things going there: so maybe we get the travel orders for somebody who went there or the Hotel receipts from an archive of people who stayed in the hotel at that time. We can bring together all of that information to tell a richer and richer story about the Yalta Conference.

## Slide 10

So what does the CIDOC-CRM do for us then? Well, it captures the underlying semantics, the underlying rules of the documentation structures that we use in the different parts of the discipline. What we did was <that> we went out and we gathered together all of the different documentation structures which were in use, and we analysed them to understand what it is that’s common across all of them. It was those common things that we then extracted and made into the objects that the CRM deals with. And we've taken them out and put them into a formal ontology. What's an ontology? It’s a much abused and misused word – ontologies are formalised knowledge, they’re the set of concepts and relationships which tell us about the possible states of affairs that exist in the domain we’re interested in. So it’s all of the different pieces that we talk about in our domains and the rules which allow them to be related to each other. What can a person do? What can an actor do? What's the difference between an actor and a person? All of those rules are things that we encapsulate in the ontology, so that we understand what these concepts are and how they relate to each other - what ways can we relate them to each other. That’s what the ontology does for us. The nice thing about them is that you and I can read them because they're human readable, but machines / computers can process them for us, so that we can get all of the boring stuff - sifting through data - done by a machine, and just look at the interesting results. This enables us to do data exchange between organisations, between different databases within an organisation. There are examples of the CRM being used in organisations where they have many, many data structures within the organisation, and they want to try and link them together. At the moment, they do that by re-keying the data into each of their data repositories, which is a very expensive and time-consuming task. Now what they can do is use the CRM to exchange that data automatically between their different research databases, and not have to re-key all of the data. So that gives us data exchange <and> data integration: we can bring together disparate data sources and combine them into a single stream of data, which will hopefully tell us more stories. We can also use it as a model for doing query mediation so that we can ask questions at the CRM level and they can be translated into queries that will attach to different databases and different data streams that we want to bring together. That's what the CRM does for us, and this gives us semantic interoperability, by giving us an extensible ontology of relationships: we have these classes, these object types that we have distilled from all of the different data structures that we use in cultural heritage, and we understand the relationships between them. However, we have produced something that sits at the top level. For a particular research question, there may be more detail beneath, so one of the things we've done with the CRM is make it extensible - we can add more detail where more detail exists within people's particular research databases. So where you have greater detail than the CRM allows for, you want to differentiate between very fine differences within your data - very fine differences within the relationships within your data - you can extend the CRM to give you that power, that additional inferential power about the very detailed stuff that you record in your research database. But by allowing it to be an extension of the CRM, by adding in the extra pieces that allow us to make extensions built into the standard, we can still query these extra pieces of information using the concepts which are available in the CRM. So when you extend, you have to follow certain rules. The rules are that everything you build must be a subclass of something that we already have, and any of the relationships that you have between things must be sub-properties of the properties we already have. Now that might sound a bit complicated at the moment, but as we go through you’ll see how these hierarchies work. You’ll be able to understand that this is a very powerful mechanism, which allows us to query using the generalised formats that we have for working the CRM, while still allowing you to go into the greatest detail that you need to for your particular research question. By doing that, it gives us this fantastically rich environment to play with, and we drive the whole thing by using events, so we are what's called an event-centric model: everything revolves around events that are in our data. Now many of our data structures don't explicitly talk about events – so for instance an archaeological database will have a record for a context, something that was excavated by an archaeologist. That archaeological context record will not have a field for “excavated by” and it won't have a field for “deposited by”, where “by” means the event that did it - the event is implicit in the data structure. We know that somebody dug it up because there is a record for it in the database, we don't have an explicit event which says: “this is being dug up,” and we don't have an explicit event saying: “this is being deposited by some social process in the past,” so those events are *hidden* in the data, they are implicit in our data structures. We get that a lot, for instance if you look at a Fine Art database, and we have a statue, there is no explicit representation of the event of the making of the statue. There might be a *date* associated with the statue, but the statue <itself> doesn't have a date, it’s when the statue was made that has the date. There might be an artist: well, the artist isn't in the statue, they’re not part of it, they weren't welded into the inside of it as part of making it. They were the person that performed the act of creating it, so we have the making of the statue at a particular date by a particular person, but we don't have a record for that event itself: it is implicit in the data structure that we know that the statue exists, so it was made, so there was a making event. We attach these other pieces of information to the statute rather than to the event, which was the making of the statue. We disguise that event. So what we do is we bring those events out of the underlying data structures and make them *explicit*, because that gives us the power to hook other data from other data streams into that event. It provides us with the hooks that allow us to do integration. By doing this, it gives us the ability to create shared explanations. Now those shared explanations are shared not only amongst people within the sub-domain where we started, but also with other sub-domains, because the CRM is multidisciplinary: it's across domain data structure. Consequently, when we explicitly extract information from a particular research database and put it into the CRM, we make it available for people who don't work in that particular subfield, but we give it to them so that they can use it as a shared explanations as well. This is far more powerful than prescribing a common data structure. We tried that and, as I said, it doesn't work, because the requirements of all the different sub-disciplines, of all the different research projects, of all the different research questions that all those different research project want to ask is *too much*, it overloads the database structure so much that it becomes unusable, too rich. But by producing this shared explanation, we can allow people to share their information across sub-disciplines, across their research questions, across their research projects. The other problem with producing a prescriptive data structure is that it tends to fossilise research. If it's taken us years to produce the data structure that we’re all going to share and use and be happy about - happy being gritted teeth happiness rather than a full smiley happiness - we can't add to that easily, because adding a new field to help answer a new research question is a huge process. It tends to fossilise things because it's so monolithically difficult to push forward, to have new ideas, to go and do something different that nobody does it. Or if they do, they don’t share, it just disappears: effectively, the knowledge that is gained from that research becomes hidden. We don't want that to happen - the only people who can make new research are the researchers who go out and have new ideas. By having those new ideas they think of new questions to ask, and they want to gather new data to help them answer those questions. Some of those will be dead ends, but we don't know whether they're dead ends until people have actually done the research, gathered the data and tried to use it to answer the questions. And of course, if it fails it doesn't matter: we've added to our knowledge that that is not something that we should bother with anymore. However if it succeeds, we've opened up a whole new area, a whole new arena for us to do research in. Consequently, we want to have dynamic question asking, dynamic processes for developing new data that we want to gather. We don't want a prescriptive recording methodology which stifles that kind of development, so by using this shared explanation approach, this object-oriented approach, where we can extend the data structures when we need to, to add new detail, we stop data being locked. Consequently, we enable research to grow, develop and morph, become exciting again. *That's* what we want to achieve, and the CRM is this ontology that allows us to do it. Now the nice thing about the ontology is that it's a language that can be shared by the IT community, by the software developers, and by the museums, libraries and archives community - the people, the experts in our domain. Consequently, we use the language as a way of allowing these two completely different groups with very different desires in the world to talk to each other, so it produces a language of interface between the software developers, the IT world and the cultural heritage domain professionals. Obviously for that to work we had to develop it using both communities, we couldn't just develop it as cultural heritage specialists, because we wouldn't have had the language that is required to talk to the IT people. We’ve seen many mistakes in the past, where the IT community have come along and told the cultural heritage community: “This is the way it's going to be!”

My all-time favourite story of this was when I worked for an archaeological unit in Scotland some years ago, and the guy who’d been the previous software developer there had gone to one of the excavation teams on the first day of the excavation and said: “Right, how many contexts are you going to find?” And the archaeologist said: “I don't understand the question.” And the computer person said: “Well, I want to know how many contexts you’re going to find, so I can build the database ready for them to all be put in. And the archaeologist went: “You’re mad - I don't know how many contexts I’m going to find.” And the computer person went: “How can you not know? You're doing science, you’re supposed to know these things!” and it was just like these two people completely missed each other – <there was a> complete misunderstanding of what each side wanted to achieve.

So what we've done is we've done the interdisciplinary work – we’ve had computer science people involved in the development so that we’re using the correct language, so we can explain it to them. But we've had the cultural heritage people – museums, library and archives community involved, so that they can talk in a language which makes sense to them. So it's a multidisciplinary, interdisciplinary approach and the hosting by CIDOC of the whole process has meant that it has actually succeeded: that we do have a standard where the computer IT community can understand what we're saying and the heritage community can understand what it is that's in there, so that it is a language that allows us to communicate between these two different communities.

## Slide 11

So what did we end up with? Well, we’ve got the conceptual reference model hosted by the International Council of Museums through CIDOC. It’s 86 classes and 137 properties - so there’s 86 types of things that we've identified and 137 different ways that they relate to each other. Now that’s not very big. Some of the chemistry ontologies – for very small pieces of organic chemistry, or for drug interactions in the pharmacy world - are hundreds of thousands of classes, with hundreds of even millions of properties relating to each other. So this is very compact, and those other ontologies covered very small things, but we cover a huge array of information. We've got all of museums, libraries and archives across all of cultural heritage - so a massive domain covered by something that is very tight. And as you get used to using the CRM, you'll find that most of those 86 classes / 137 properties will stick in your mind, and you’ll be able to remember that there’s one there. You might not know precisely what its definition is at any one time, but you'll know that it's there and that you need to go and look it up. It's very human in scale, so you can actually remember it and use it easily. It takes a little bit of practice, but 10 days with and you will find that you have most of it in your head. It has the capacity to explain *hundreds* of metadata formats. Now when we say “explain hundreds”, we really have done it: there was one project in the research library group which mapped 1460 different data structures using the CRM together. So we have literally thousands of data structures that have been mapped to the CRM, which means that we can explain all of the contents of those different data structures using the CRM: we've made them interoperable, we've made them able to have a shared explanation of the world. It was accepted by at the International Standards Organisation, ISO Technical Committee 46 in September 2000, and it went through the fast track ISO process, which means that it actually became a standard in October 2006. That’s the fast track - six years – imagine what the slow track is like! We got it through the process and we have ISO 21127, published in 2006. That’s the international standard, so it's not just something that’s thought to be a good idea by a community within. We’ve also got it accepted excepted at the International Standards Organisation level, so that many different countries looked at it and voted on it and got it and it’s been accepted. It’s now something that is a touchstone across the international scene, as well as within our discipline. What does it do for us? Well, it serves as an intellectual guide to creating schemata formats and profiles, so when you go out to design your database you can use the CRM as a way of checking that you not doing something silly. As an example of this, the Finnish National Gallery redeveloped their collections management system and they used the CRM to help them do that. They had five people working on the project, and they estimate that it saved them one year’s development, so that instead of it taking nearly 2 years to develop it, they completed the redevelopment of their management system in just over 10 months - five man years of work was saved by using the CRM. They found it was so useful to them that they were able not to worry about things - they could use the CRM to say: “Yes, that’s the way we do it!”, because hundreds of experts have spent years giving us that information. In effect, using the CRM to help you check the data structures that you're about to deploy into your new project saves you about 150 person years of thinking, because that is roughly what's gone into the development of the CRM. We've worked on it for more than 15 years, and there have been literally a hundred or more people involved, so 150 person-years - and it’s free to download! That’s pretty cheap really. Lots of research, very little time for you to absorb it. So that’s one way we can use it. It's also a language for us to analyse our existing sources so that we can do the integration tasks. We can bring things together and get a nice rich web of information out of our different data structures and different data sources, to bring everything together. It gives us the language for taking these different data structures and understanding what's actually in them,<for> making explicit the events – taking them apart, opening them up so that we can combine our different data sources, different data streams into one big richer environment. Finally, it also acts as a transportation format, by labelling things correctly with the CRM classes and the properties that link the objects together, we can produce an environment that allows us to move data about, where all of the meaning is in the data. I've got everything I need to explain encapsulated in it, so it gives me an environment where I can do integration -which we've talked about - but I can also migrate from one generation of software to the other, and I can provide for digital preservation of my data over long periods of time, because it's a very simple format to hold it in. We can hold it as text, but that text holds all the richness of the semantics of all of the rules we have embedded in our desk instructions, in our ways of working. Obviously, these types of things work very well in an Internet environment, so we can push that data out, and other people can use it, they can harvest our data and start to use it for their applications. They can enrich their knowledge and their environment to give them a much better approach to tackling their questions. It gives them more power.

## Slide 12

We can think of the CRM as being a piece in the chain of our intellectual understanding of the analogue world that we live in. So here's the world - with tigers and telephones and desks and all that - in a nice analogue way, with electrons flying around and interacting with each other, and we abstract from that using some conceptualisation: the things that we're interested in. We go into this analogue world and we say: “Yes, we’ve got this thing here and it’s a tiger, and we’ve got this thing here and it's a snake, and we’ve got this thing here, which is a strange person who lives in the jungle with a telephone.” And we have a telephone. We then build data structures and metadata to tell us where things are in there, and put the data into databases which refer to that. What the CRM does is it sits and informs this conceptualisation, and motivates the data structures that we actually produce. So it acts as a link between us thinking: “Ooh, there’s a tiger!” and “Right, we need a database where we’re going to put the tiger. We need a data structure which will allow us to talk about tigers and the things that tigers do.” Obviously, waiting for the telephone in this case! So how do we actually set about that? Well, we use the CRM to provide us with a touchstone that says: “These are the kinds of things that we're interested in, and these are the ways they relate to each other, these are the properties that link these things together.” So it enables us to take the raw conceptualisation and put it into data structures and metadata structures which will then refer back to the real world, but in a way that we can follow, that we can reuse.

## Slide 13

How have we actually built the CRM? What has it actually done? Well, it’s a formal ontology, in that it’s a set of rules - the set of things that we’re going to talk about and the relationships between them - and we've defined it in a language called TELOS. Do you need to know TELOS in order to work with the CRM? Absolutely not, it's just a language that allows us to define these things in a sensible manner and automatically process them. Nobody really understands TELOS in the cultural heritage community – it’s a thing used by computer scientists. So we don't need to worry about the fact that the CRM is in TELOS or in anything else. But TELOS is a useful way of holding these ontologies. The actual data, the instances of the CRM, as it’s referred to, can be encoded in many ways. We can deploy this knowledge about how things relate to each other -this ontology - in lots of different ways. We can use a relational database – we’ve seen deployments of this using Oracle, for instance. We can use object-oriented databases - extensions to relational or pure 00 databases. We can use XML, and we’ve seen many deployments of CRM into XML or, probably a little more richly, RDFS. If you don't know what those are, it doesn't matter - those are what you ask your computer science buddies to deal with, and they'll understand it. The CRM will explain to them what it is that they’re going to deploy into those different technologies. So it doesn't matter if you don't know what they are - you've got the language to explain what you want to achieve in those different technologies to the person who does understand them. We use a thing called Multiple isAs. A multiple isA is a way of expressing richness of structure in a very simple way. We do this so that we can achieve uniqueness of properties. Let's take an example of me: I am a person and - simultaneously - a person is somebody who can act, who can do stuff - and I'm a biological object. Some people think maybe not, but I am a biological object, and so I am simultaneously both a biological object and an actor. And that’s what a multiple isA does for us - it allows me to have the properties of being a biological object *and* the properties of being an actor together at the same time. I’ll show you a little bit more about that in a few slides’ time. We also use the concept of multiple instantiation: for instance, when we give an object or sell an object to a museum, we're going to do multiple things. We’re going to change who looks after it: it’s going to go from me to the museum. I’m going to give the museum the custody of the object, but I'm also going to give them ownership of it, so we're changing ownership. That’s two *different* tasks that we are doing by depositing this object with the museum. Now sometimes I'll give a museum the object, but I’ll keep ownership of it – so I’m going to *lend* it to the museum. Well, that changes custody but it doesn’t change ownership. Or, *I* might continue to look after it, if it’s a big thing in a field somewhere, but the museum might take ownership of it. So <for example>, I’ve got a big steam engine in a field. I’m going to leave it in the field, I’m going to continue to look after it, but I’m going to give the ownership of it to the museum. So we separate those out into two different things - change of ownership / change of custody. Then we can combine them to produce different combinations of those depositing things with museums, which allow us to say exactly what’s happened without having to have lots of different types of giving something to a museum. It also allows us to deal with things like something being destroyed deliberately: I break something or smash it up, or it just gets broken. Entropy takes over and it becomes a pile of bits. In either of those cases, the thing is destroyed, but in one of them I’m doing it: it’s an activity, I’m actually there breaking it, <but> in another there isn’t an activity, there’s nobody involved with it, it just happens. So I don't want to have a destruction which is done by somebody, and another case where the destruction is *not* done by somebody. I just want to be able to combine the different aspects of things – “the destruction of it” and “there’s somebody doing it” - by using multiple instantiation to bring together those different types of thing into one. So I can have a destruction event and the activity of somebody doing it, and by combining them I can now say *who* did it. Multiple instantiation greatly simplifies the picture of the way the world works. We can use this in a number of ways: I can talk about those kinds of events where things get destroyed, and I can also talk about events where we give something to a museum and it involves us moving it – I’m going to give the museum the object, they’re going to take custody of it, and I’m going to physically move it from my house to the museum. So now, that one event is three different things happening. <With> all of the possible combinations of those it starts to get very complicated, and if we didn't use multiple instantiation we’d have a model which was very, very complex. That's precisely what the relational models all did, and that's why they became so phenomenally complex. By breaking it down into simple things which we can combine together to produce the more complex cases, we don't have to produce an incredibly complex data structure with lots and lots of rules around it. We just bolt together the pieces that allow us to tell the story we want to tell. While we have multiple isAs for classes – for the things that we’re interested in, for the properties that link them together we also have an isA hierarchy - I’ll show you that for some of them a little later. This allows us to capture different levels of abstraction, so I can talk about somebody / something being involved in an event or an activity, and then I can go all the way down in very great detail to: “This person was involved in this activity in this role.” So I have different levels of complexity, but if I want to do a general search, I want to know about *all* the different ways that somebody could be related to this event or activity. So instead of having to make very complex queries I can have an isA hierarchy that says if you just want a general relationship between an activity and a person you can query at this level, and it will include all these other properties which link them together in special ways. So multiple isA for properties gives us the same power about how things relate to each other as we use for the things themselves. <As for> methodological aspects, we only introduce an entity or a class where they are anchors for properties, where they are structurally useful. So we don't have lots of things that we could think of if we were having a really good day thinking about things we don't see in the real world, which aren't useful for us to hang actual properties and relationships between classes on, for the things that we actually seen in our data structures. Mostly, this stuff is built from the bottom up - we've gone and looked at all these different data structures, and extracted from them the things that are common. This is what we actually do, and we’ve added to it a few things that give us structure, which allow us to combine things where we needed them in order to hold the stuff that the different data structures actually have. So we've linked things together that way. But in the real world we often see shortcuts - quick ways of getting from an object to something, for instance. Let's take an example: when we look at this object we can say that it's “so long”. In my database I will have a field that says “length”, and it's 6 cm long. Now what we’re actually saying is that there was an event where somebody measured this object. In that case it was me, and that measurement was of the object, and produced the dimension “length of Stephen's clicker” and it gave it a value of 6 cm. I can take a shortcut which goes from the clicker straight to the dimension - 6 cm - or I can say there was a measurement event in which Stephen measured it using this technique, and produced a dimension of its length, which was 6cm. The shortcut allows me to represent what we actually see in data structures without producing the events, where that would introduce too much complexity. However we can always add the event, so if later on I found a notebook about all of the things that I've measured, the techniques, when I did them and what measuring stick I used and so on, I could then incorporate that information into the rest of my data by using the join that is the event of me measuring. So I can add in all of those extra pieces of information, and that's my hook, the event is the hook. These joins allow us to represent what we actual see in data structures, but give us the ability to do the long way round without any problems. It’s there as well, and the shortcut tells us that the long way round is there, exactly what it is and the definition of the shortcut, so that I can do the long way round and add in *other* data flows, other data streams to give me my richer environment over time.

## Slide 14

Let's now look at how we use multiple inheritance in order to remove complexity. This is one of the things we said we did in the previous slide, so why do we do it? Let's take the more traditional single inheritance: here I’ve got a museum artefact, and I’ve got a field for the museum number, for the collection and for the material. And then I’m going to have canisters: things which we put stuff in. We’ve got the base and the lid to sit on top of it. We’ve got information about the number of the container and the number of the lid. Then we have a special type of museum artefact for ecclesiastical items, which belong to a particular church. So we have a field telling us which church they belong to, and then we have a special type of ecclesiastical item called a holy breadbasket. Holy breadbaskets have a base and a lid - a container and a lid. So I have a field for the container and a field for the lid. Now this means that the container and lid fields appear in two different places. If I want to find all of the objects which have containers and lids, I have to know to go and search in two different places. Obviously, as my types of objects grow, I will have to go and look in multiple places. I may have to go look in 20, 30 or 40 places, and the queries that I therefore execute become very complicated. Now let’s take the multiple inheritance form the way we do it in the CRM. Here we've got the same thing – we’ve got museum artefacts, which have museum numbers, collections and materials. Then we have the concept of “canister”, which has a container and a lid, and we have the concept of an ecclesiastical item, which belongs to a particular church. But now, instead of “holy breadbasket” being an ecclesiastical item which has “container and lid”, I make holy breadbasket simultaneously “canisters” and “ecclesiastical items”, and that means that they inherit the properties of both “canisters” and “ecclesiastical items”. So my holy breadbasket belongs to a church *and* has a container and lid. Now this greatly simplifies things; because I am using *these* container and lid properties, I can get at holy breadbaskets by searching for the general “canister” properties. And this means that as my data structure gets richer - and I have lots of different possibilities, I can move an object to a museum, give the museum the object, or let the museum look after the object (“change of custody”/ “change of ownership” and a “move”) - I don't have to have all the particular cases. I don't have to have “it moved to the museum but I didn't give them ownership” / “it moved to the museum and I still come in and look after it” - which is kind of unusual, but we could conceivably have that situation. I don't have to have all those special cases as different pieces in the data structure. I just have the concept of moving it, of acquiring ownership and of acquiring custody. Then I can add them together in whatever combination I need to give me the actual circumstances behind this particular case. So this allows me to greatly simplify things. However, it does mean that some object-oriented programmers get kind of nervous, because this is behaviour that they don't like. It is not traditionally used in object-oriented programming, but it’s perfectly acceptable in object-oriented modelling. So if somebody says: “Oh, you can't do that in object orientation,” what they mean is that you can't do that if you're an object-oriented programmer. So you can just gently persuade them that you're modelling, not programming, and consequently this is permissible behaviour.

## Slide 15

Let’s have a look at an example of some data: this is laid out using XML and a simple access XSLT transform, which takes the base data and makes it look pretty, and we can automatically process this. As you can see as I go through it, we can actually read it as well, as humans, so we've got the ability to read the data *and* process the data from the same format, which is very handy. We don't have to translate it for us to read and for the computer to read. This is an example actually extracted from a document - a lament cloth - called Epitaphios 34604, which is at the Benaki in Athens. So here we have this object, which is a man-made object. It was a given to the Benaki - custody was transferred, and it changed ownership through this transfer event, which is simultaneously a transfer of custody and an acquisition event, so this is an example of that multiple instantiation we’ve been looking at. The custody was surrendered by the Metropolitan Churchly of the Greek Community of Ankara. So this is part of the exchange of populations that happened in the 1920s between Turkey and Greece, when all the Greeks in Turkey came back to Greece, and all the Turks in Greece went back to Turkey. The community objects were exchanged as well, so there was a Metropolitan Church church in Ankara, and all of the objects that belonged to that church were shipped back to Greece. They were put into the Benaki Museum and other museums, and the whole thing was overseen by a foundation called the Exchangeable Fund of Refugees, which was a government organisation that handled all of this. The custody used to belong to the Metropolitan Church, and it was given to the Benaki - they were going to look after this object for everybody. Now the owners of it used to be the Greek Community - the Metropolitan Church of the Greek Community of Ankara - but it was now going to be owned by the Exchangeable Fund of Refugees, this national foundation. Later on, the Exchangeable Fund of Refugees was dissolved, and the ownership actually transferred to the Benaki, but that was much later. So the original transfer – the title - was transferred from the church to the Exchangeable Fund of Refugees, which was a national foundation. Now this action, this transfer of the cloth, was carried out by the Exchangeable Fund of Refugees at some time between 1923 and 1928. Here is the time span for the transfer of this object, and it occurred some time between 1923 and 1928, so this is the time primitive that we used to actually hook to time. It took place in Greece, that which is a nation, a republic that falls within Europe, and Europe is a continent. This is information we can extract from the TGN, so here we’re showing how we blended two different data streams to show a fuller set of information. That allows us to get at what Greece is, in case somebody didn't know - we can expand on our knowledge by bringing in information from another stream. Now you can see that this is *pretty* readable – it’s a little stilted in its English, but we’ve chosen the names of the objects and the names of the properties so that we can read them reasonably well. They’re not perfect under all circumstances, but most of the time it reads as a sort of fairly formalised English. In addition, because we are able to explain what all of these labels are to the computer with a simple document, the computer can process this. It's both machine-readable and human readable in one format, which is very nice, it’s a nice elegant solution.

## Slide 16

What does the CRM consist of? Well, it consists of these top level classes. This is the very upper level of the CRM, and if you understand this basic diagram, you understand the CRM overall. It's pretty straightforward: at the core of it are these temporal entities. These are events, things happening in the past, and these are where we join together everything, so we are an event-centric model, and this is the top of the event-centric subclasses, the very top of all of our events. It provides us with the *only* link to time - we don't allow objects or people to be linked to time, we only allow events to be linked to time, and people or objects or places to be linked to those events. So events happen within some piece of time, and they happen at places. Places are interesting, and we’ll talk about them again in a little while, but places are *not* something you can take a photograph of: a place is the mathematical construct of where something is. The inside of my wedding ring is a place – it’s a mathematical shape we can describe. It’s *not* somewhere on the face of the Earth necessarily. It *might* be, but it could be the front of a ship, or the top of this table, or the front of a painting, or the back of a coin. All those things are places and they’re mathematical descriptions of where it is, *not* the stuff that’s there at a particular point in time. We’ll come back to that and I’ll show you some more detail about it in a few minutes. So the thing happens, the event happens at a point in time and at a place, and we can have actors taking part in it. Now actors are anything which we can consider to have legal responsibility for what they do: individuals – you and me – we are actors, but also groups of people. Whether they are a formal group of people, like a company, or an institution like FORTH, or whether they’re informal groups of people – a “school” of artists, for instance - we could consider a group. So actors are individuals or groups of individuals who can do stuff together. They affect *things*, and we can divide the world up into two types of thing: physical things that you can smash windows with, and conceptual objects - those are things that are the products of the human mind. The big difference between them is that physical things can be destroyed. By destroyed I don't mean that they cease to exist and they become energy, because that would mean there would be a fairly big bang every time something got broken. What we mean is that they're taken out of our sphere of interest, so we no longer want to document them. If I take a desk and I break off its legs and make it into little pieces of wood, and I'm not going to record what happened to those little pieces of wood, then it's been destroyed and it’s no longer part of our domain of documentation. However, if I take something apart into its component pieces and then I keep track of them – I take the legs off and I start tracking the legs - they've not been destroyed. The table has not been destroyed in the sense we talked about, it has been transformed into some other things. We’ll talk a little bit more about that in a little while. Now, conceptual objects can’t be destroyed, because they’re a product of the human mind, an idea. Consequently, we can't destroy them – all we can do is destroy the last carrier of them. If I’ve got them written down in books and I burn all the books, so nobody can remember what was written in the book, then we've lost a conceptual object, but if we found another carrier with it on, then it’s suddenly back. The carriers don't have to be books or CD-ROMs or floppy disks or computer drives; they can be the human mind. The CRM, for instance, is held in many people's minds and on many hard drives, and many copies of it have been printed out, and it's on CD-ROMs and floppy disks all over the world. So in order to destroy the CRM I'd have to gather together all of these copies and printouts and burn them all. I’d have to destroy everybody’s hard drive and all of the CD-ROMs that it's been put on, and then I'd have to go and shoot everybody who could remember it. So I’d have to be shot, of course - and now, because you've seen this diagram and you can remember it, I’d have to come out and shoot all of you as well, which is a bit drastic really. So I think the CRM is going to survive for a while. Conceptual objects can only be destroyed by losing the last carrier of them; physical objects can be destroyed by making them into small enough pieces that we’re no longer interested in documenting them. That’s it basically - we've got events, we've got actors, we’ve got places, time spans and the things that we act upon - the physical objects and the conceptual objects that we act upon. That’s the base structure of it, but we have a couple other things that we like to do. Humans tend to like to give things names, so you give your pet dog a name, you don’t just refer to it as “dog”- you give it a name. And we are constantly making names for things. Those names themselves are things that we can study the history of. So what we provide for is the ability to name absolutely everything *and* keep the name as an object itself, so that we can study the history of the use of the name, its derivation, how we’ve recorded it and how it's been used over time as a subject in itself. We think it’s very important to differentiate between the *thing* and its name: I'm not Stephen, I’m a *thing* which is called Stephen. We differentiate that very core – it’s a very central theme to the whole of the CRM - we differentiate between the things, their identity and their names, because I can have many names: I can be Stephen; I can be Stephen Stead; I can be Steve Stead; I can be S. D. Stead; S. Stead and so on. All of those are names that apply to me, and other people can use those names, so for instance my grandmother is also S. Stead - Susan Stead. So the same name can be used by different objects and we can study the use of the name independent of the things that have that name applied to them. Appellations are something which are worthy of study themselves, but can be applied to everything that we have. That’s one overarching ability that we have. In addition, as scientists we tend to classify things: one of the driving forces behind science is that we want to organise things in some way and classify them, and we provide for that by allowing us to type everything, so we can add a type to every type of object that we have in the CRM: we can give things their classes. Classes or types can be applied on a one-to-one basis, but many people might classify a particular thing in different ways, so I might have many different classification systems from different organisations. Different research questions might classify something in different ways, so we allow types to be applied to everything multiple times. I can have as many different classification systems simultaneously classifying one object as I want, and this gives me a great deal of power. In addition to that, the types themselves are conceptual objects, so this allows me to have “types of types”, and “types of types of types” if want. I can be as deep as I want to be, but it allows me to discuss how the types have been derived, how they’ve been documented, how they've been used over time. In the same data structure that I used to discuss the objects I'm interested in, I can discuss how I’ve classified them and named them over time. So the names, the types and the objects all sit in the same data structure. That means that when we migrate, when we lift our data out and move it to a new system or put it into a digital repository to preserve it over a long period of time, I can use the same data structure to encapsulate the data *and* all of the stuff that I used in order to classify my data and name it.

Let me give you an example of how this is useful: in the 1970s, there was a big project in Southampton in the UK to work on a collection of mediaeval pottery. This mediaeval pottery was classified in a great deal of detail and measured very accurately, using a variety of measures: the size of the shards, what percentage of the vessel was there and a whole range of other very detailed pottery research. All of this information was stored on the Southampton City Council mainframe, their big computer, and it was marked as being of vital importance that this data was preserved. But space on the mainframe was very expensive, storage space on the mainframe and processing space on the mainframe was very expensive, so everything was very heavily coded. There were codes for the fabric, codes for the form, codes for which part of the vessel the shard was from. Then all of the different measurements were put into fields which were just the right size, and they were strung together into a long string for all the different shards of pottery. All of that was loaded up on the mainframe and the initial research was done with it. The initial research was completed and they were going to maintain this data. Religiously over 20 years, 25 years, they kept the data. The mainframe migrated from one version of its operating system to another, they very carefully migrated the data and checked that they could read it. Every year the tapes were exercised, spun off and spun back on so that they could check that they could read the data off the tapes, and it was all maintained *religiously*, so after 25 years we could still get all of the data.

Then somebody came along and said: “Oh! I know about that dataset and I want to use it, because I can do some new research with that. There are some new questions that I want to answer, and that data will have all the information I need to extend my research beyond the set of shards I'm currently working on. I can look at this other big catalogue of shards which was done in the 70s, so I’ll get the data off the mainframe.”

So off he went and asked for the data, and they loaded it up onto the mainframe and they I pulled it off for him and gave it to him.

And he said “So what do these codes mean then?”

And the guys who ran the mainframe said “We’ve no idea! That wasn't our job, we don't have a record of what the codes meant. You’re supposed to have that.”

“But I don't have that.”

So we had this beautifully preserved data and we didn't know what it meant. It was absolutely 100% accurate, it was just the same as when it went into the system, but we’ve got no idea what it means because they've lost the manual which tells them what the codes are. Scary - what we can do with the CRM is - because we can hold all of those codes, all of those desk instructions, all that information about how we’re storing information in the same data structure as the data itself - it means that when we move it we move everything. We move all the codes; we move all of the documentation with it, so that we can access the data and actually use it when we move the data. That's one of the great things about the CRM: it allows us to capture all the desk instructions, all the coding, all the naming in the data structure and tell us what it all means. So that when we get the data, we’ve got everything we need <in order> to understand it embedded in the data with it, and that makes a huge difference if we’re trying to digitally preserve things over the long term. That was a story about 20 – 25 years; if you were trying to do that over one hundred years, where there is not even a chance that you might find something who can remember it, or a thousand years, if that’s the goal of some of our digital library projects, then we have to encapsulate all of the meaning, all of the desk instructions, all of the codes into the data itself. Otherwise, as soon as it goes out of living memory, we’re dead. Because the chances of managing to keep the actual instructions, the actual terminology lists, the thesauri and the glossaries with the data – well, there’s no chance of it over long periods of time, because people will forget to do it, you’ll forget to preserve some document. Consequently, the data becomes useless, and all that effort you’ve put into preserving it is wasted. We don’t want that to happen, so the CRM provides us with the functionality to capture the data and the data around it. All of our desk instructions, all of our metadata can be embedded in the same data structure and carried around with it, which makes it of very great utility for digital preservation.

## Slide 17

Within the CRM we have different types of relationships between the classes, and we can broadly give you the types of those, though we’re actually going to deploy them in different ways for different classes within <the CRM>. Generally speaking, we have the ability to give the identification of real world items by real world names. That’s the appellations that we’ve just looked at – the E41 appellations –it just means that everything can have a name. And then we can classify everything - again this is E55 types we’ve just talked about. There is a general rule that everything we have we can classify, we can add a type to it. We also provide for part decomposition, so we can talk about the parts of it: I can talk about the actors that make up a group; I can talk about the different pieces of a table; I can talk about the different pieces of a chess set. I can also break down periods: I can talk about the Early Bronze Age and the Late Bronze Age. I can talk about parts of places: I can talk about this room being in this building, being in Heraklion, being in Crete and being in Greece, so I can break down place. And I can break down time, so I can talk about the beginning of this lecture, the middle of the lecture and the end of the lecture as being different time spans. So I can break all of this down. Now obviously <with> actors, when we break down actors we’re talking about groups of actors, we’re not talking about ripping my leg off so that we’ve got a bit of me as being a part decomposition. We don't go to quite such ridiculous extremes. We talk about the “participation of persistent items”. What’s a persistent item? A persistent item is something that I re-identify without watching it the whole time: I can look at this clicker, and if I hide it behind my back and bring it back you know it's the same clicker. So it’s a persistent item, it has identity over time, independent of me continuously observing it. That produces a useful set of properties - we can talk about things that are persistent. And these persistent items can participate in temporal events - by physically being there if it’s a physical object, or by having a carrier of it be there in the case of a conceptual object. So I can talk about the order to advance as being present during a military manoeuvre, or the Declaration of Independence being present at the celebrations of 4th July. These are things which are conceptually there, there are carriers of that idea present at the event, or physical things at the event. By using this idea we can create this notion of history as a set of world lines. I talk about the object travelling through time and space, participating in events, and it’s that participation that gives me their meeting with other persistent items, which allows me to tell stories about the involvement of things. I’ll show you some graphics of that in a few minutes. *Periods* are very different to the periods that we tend to think about, where a period is just a chunk of time. Periods in the CRM are very specific: they are a cultural, social phenomenon bounded in both space *and* time. So you can't talk about the Roman period, you have to talk about the Roman period in northern Italy, or the Roman period in southern England. You have to say where you're talking about that cultural-social phenomenon, because the cultural-social phenomenon will have different temporal bounds at different places. In southern England the Romans were there for a far different period of time than they were in northern Italy: they started much later and ended much sooner. So you have to say *where* and when a particular social-cultural milieu was present. It could be that you have multiple social-cultural milieus present in the same place at the same time. We can talk about a nomadic community being in the same spatial area as a pastoral community, where one group of people live and work the land and don't move around, and another group of people move around in the same physical area, <or a> much bigger area at the same time. But they are different periods, they are different social cultural phenomena.

We can talk about periods being located in space-time: “Where are they and when are they?” and physical objects be located in space: “Where is this clicker, where is …, where am I? Where are we doing this lecture?” We can have a loose idea of the influence of objects on activities, on products and vice-versa. I can talk about a dress being made for a wedding: the event of creating the address was in order to support a particular wedding. Or, generically, I can talk about the general class of events that something is made for. So we have this general feeling of influence between events, and objects and events. Finally, information objects can refer to anything, so I can talk about any real world item and say that this piece of documentation is about it, that this book is about this clicker - I’ve written a whole book about this clicker. This lecture - the video of this lecture - is about the CRM, so I can talk about the information objects referring to real world objects, things that we've identified in the CRM’s class system, instances of the classes.

## Slide 18

So let’s have a look at the big one - the thing that drives the model - the temporal entity hierarchy. This is all the different things that are temporal entities. Right at the top of the tree we’ve got temporal entities - E2 temporal entities - which we looked at in the middle of the previous diagram. It breaks down into two different pieces: we can talk about periods, the social-cultural phenomenon bounded in space and time, and condition-states. This is where we talk about something being in ruins between 1950 and 1975, so it’s the state of something over a period of time. Periods break further down into events, and we’ll look at what extra pieces we get here. A that period is a social-cultural phenomenon, whereas events are where we start to get things being present at, so we don't talk about the clicker being present in a period, but we start talk about it being an event. Now every event is a period; it just depends on what scale you're looking at things. A wedding could be an event or it could be a period – it’s a social-cultural phenomenon bounded in space and time, a wedding. But it could be an event, because we’re going to start talking about people being at it, objects being at it as well. Then we have a class of things which are the end of things, the end of existence: people's death, the dissolution of groups, the destruction of an object; and we have the beginning of existence of things: the formations of groups, births of individuals, the creation of products of our mind. And we can also talk about the production of something: we make something. And then we have the general class of *activities*: this is where people are doing stuff. The different things that we can do are: we can modify something or produce it - we consider production to be a special case of modification - we can move things, we can transfer custody, and acquire them - that's the transfer of ownership - curation activities for looking after collections of objects. After all, one of the main movers in the creation of this has been being museums, so obviously we’re interested in how we curate museum collections, and we have a special activity for that. Joining and leaving groups: I can have a group of people and new people can join it and people can leave, so those are activities that people take part. And we have this general class of attribute assignment, where we say: “This is where I said that this is a clicker, so I classified this as a clicker, I gave it an E55 type and I can say who said that, when they said it, using what criteria.” That's the general case of attribute assignment, but we have special cases. <There is> *type assignment* – I’ve just classified that, it’s a special case. We have a special class for that. <There is> *condition assessment*, where we assess a condition-state. *Identifier assignment*, where I give this a number, so somebody gave this a number: this is item number IOWWZ9005 - that's the serial number given to it by the manufacturer, so they did an identifier assignment. Typically, identifier assignments are when we absorb the object into a museum. When it’s absorbed into the museum we will give it an identifier that museum is going to use for it. Objects will, over their life, have many identifiers assigned to them. One one of the important things to remember to do in any information system is never discard identifiers, because somewhere or other somebody will have used that identifier to identify the object. If you no longer have a record of <the fact that> that it used to be called 27 and now it's called 72, then anybody who referred to it under its old name will not be able to get out that data again. So you have to keep a record of all of the identifiers. And identifier assignment allows us to say when things happened and who assigned the identity: we can look at the history of the identification of the different numbers and letter strings that have been applied to an object over time, so that we know which identifier was used by which person at which time. And then there’s the *measurement event* – when we were talking about the length of this clicker earlier on, we were using an E16 measurement activity in order to take that measurement. Now there are a couple of interesting things here: E11 *modification*,which is a superclass of production - production is the beginning of the existence of something and it's also an isA modification - modification and transformation have very similar characteristics, but there is a key difference. One of the key things that we need to look at here is the difference between modification and transformation. In a modification we change the matter of something, but we don't change its identity. Here, for instance, we take a laptop computer and we change the keyboard on it. We’ve changed its matter and it’s now made up of different components, but it's still the same laptop, it’s still Stephen's laptop. Whereas <with> transformation we don't change the matter, we just change its identity. So, for instance, when Tutankhamenis transferred from being Tutankhamen to being a mummy, it’s the same physical matter that makes up his body, but we change his identity from being Tutankhamen to being Tutankhamen’s mummy. There are not very many cases of that outside the architectural world, but architectural historians do that a lot. For instance, a building will be built as a windmill and then changed into a house and maybe then changed into a studio for an artist. And then it might be changed back to being a windmill. Or a building could be built as a hotel and then changed to being a private residence, and then changed back to a hotel, and then changed to a shop. So architectural historians are constantly transforming things, where the matter - the building itself - stays the same, but we change its identity because we want to look at it in a different way. Transformation is maintenance of the physical stuff that make it up, but changing its identity; modification is changing the matter and *not* changing the identity. There we're changing a piece of it, but it's still the same thing that we are going to document.

## Slide 19

That’s the hierarchy for temporal entities - let's move on and look at *what* a temporal entity is. All the way through the standard, we identify these classes and we say what they are. And it's not what you think the word means that matters, it’s what the scope note says it is. For each one of the classes we have a scope note, and this is a copy of the scope note for E10 temporal entity, so it says: “This class comprises all phenomena such as the instances of E4 (periods) and E5 (events and states) which happen over a limited extent in time. In some contexts these are also called *perdurance*. This class is disjoint from E77 (persistent item). This is an abstract class and has no direct instances. E2 (temporal entity) is specialized into E4 (period), which applies to a particular geographic area defined with a greater or lesser degree of precision, and E3 (condition state), which applies to instances of E18 (physical thing). That is the definition of what it is - it doesn't matter what you think it is, *that* is the definition. So whenever you read a class name, you must know what the scope note is to know what we mean by that class. The upshot of this is that it’s limited in time; it’s our only link to time, it spreads out over a place or object and is the core of the model but these are just our interpretations, this thing is the text that tells us what an E2 temporal entity is. Every single class has a scope note, and that's what's in the standard. You must read those to understand what it is we meant. Every property *also* has a scope note, so everything that we have in the whole standard has these scope notes, which precisely tell you what we meant. When we do translations of the CRM into different languages - at the time of shooting this video there were instances of translations into Portuguese, German, Japanese, Korean, French, I lose track - lots - each has translated the scope note, but it doesn't translate the label. E2 (temporal entity) is E2 (temporal entity) whether you're reading this in Portuguese, French or German. The scope note is changed into the language to try and explain it to an audience who wants to read in that language, so these names are labels, they don't mean anything. It's the scope note that conveys all the meaning.

## Slide 20

<Moving on to> subclasses of temporal entity, we’ve got “period”, which binds together related phenomenon, the social-cultural phenomenon bounded in space and time. It’s where we can do inclusions, so this is where we add parts. Temporal entities don't have parts, they don’t have the beginning of the temporal entity and in the end of it, it’s only when we get down to *periods* that we can talk about the beginning and the end. It’s confined in space and time - bounded in space and time - and it's our basic unit for temporal-spatial reasoning. This is how we do temporal-spatial reasoning: “This happened before that; it happened next to it; before it, next to it and so on.”

*Events*, which are below period, start to look at inputs and outcome. What went in? What was the output of the event? This allows us to talk about participation in the most general sense of people and the presence of things. Things are actually at the event. It gives us a weak causal reasoning, so we can say: “This caused this” in a very general weak sense. And you can think of each *event* actually being a *period* if you're looking at in sufficient scale. So a wedding is a period if you want to detail every little thing that happened at the wedding. You would look at that, <so that> you could conceive of the wedding as a period, and then each of the individual pieces of it as being events.

Then below *event* we have *activities*, and this is where you start to add the intention: “I wanted to do this, because…” *Influence*: “This activity influenced that activity” and *purpose*: “This was in order to do something.” It <also> adds the use of tools, so we can start to say: “This design or procedure was used, this hammer was used, or generically hammers were used to do this. Or moulds were used to make something.” These are the types of things that we add in at these different levels - each time we go down a level in the tree, we add some new capabilities, new stuff. *Activities* inherit all of the things from *periods*, so here we can talk about inclusion: activities have the ability to be broken down into different pieces. <For examples,> we can talk about the making of this video: it had at the beginning some preparation, then we've had the activity of actually making the video - me standing here talking - and then there’ll be some post-processing, where the soundtracks will be cleaned up and the different tracks of film will be edited together to make the final product that comes out to you. We can do an activity of making the video, or we can talk about those individual pieces of making the video and bolt them together. *Activities*, because they are a subclass of *event*, which is a subclass of *period*, gain the qualities that we have given to period. The properties that allow us to do the inclusion topologies *come down* to activity.

## Slide 21

Here are the main properties that we use to express those general things we’ve just talked about. An E2 temporal entity has time span, which is our link to time. Then the periods: we get *place* – “took place at” a place; *consists of* “a period”, which is where we can break it down into individual pieces, and *falls within*. There is a subtle difference between *consists of* and *falls within*. *Consists of* talks about things which are part of it - the Early Bronze Age, the Middle Bronze Age and the Late Bronze Age all form part of the Bronze Age, whereas *falls within* is a more generic thing - the 20th century contains my birthday, the date of my birth, but it's not a necessary part of the 20th century to have my birthday in it. We can talk about that looser relationship - *falls within* - or we can break it down into parts. Both of those capabilities are brought in with *period*.

The events give us *participation*: *participant*, *actor* and *occurred in the presence* *of* are general relationships to persistent items. Remember that persistent items are those things that we can look away from and look back, and know it is the same thing, and they apply to both physical items and conceptual items. I know that it's the same poem, even though I haven't watched it the whole time. Persistent items are both material and immaterial things. Then we go down to *activity*, and now we can say that instead of just being generally there and involved, they actually did it, so this is *carried out by*. We also have the ability to say what role they had, so we have an additional property of a property, which says in what role they did. And then we have *had specific purpose*, so that I can talk about something in preparation for something else. <For instance,> I can talk about making a wedding dress was in preparation for the wedding of somebody, a particular wedding. Or I can have general purposes, so I can say: “The making of this wedding dress was to support weddings in general. I don't know which one it was for, it was just made to be used at a wedding.” Or: “This object was made to be used in a particular ceremony. I don't know which instance of that ceremony it actually occurred in. This is to a particular event, and this is to the general type of events that we can go to.”

## Slide 22

We talked about the fact that there's an isA hierarchy for properties, and this is the isA hierarchy for the general participation. Right over here we have the most general form: *occurred in the presence of*. That’s a persistent item *was present at an event*, so if you remember we go back to here and *occurred in the presence* *of* (B12) is a persistent item at an event. This is that is that top level property relating those two. Then we can go down into more detail, and we can talk about *moving something*. Well obviously if I moved something, the thing had to be there. I can use a specific object and I can use specific techniques. These specific objects were used as a constituent, so I can use something as part of the thing I’m making, for instance. I can use this fork to be made into a bracelet - I can do use things to make parts of other things. I can *have participation* and then, in more specific cases, I can *leave a group separated from* or I can *be the mother of* somebody. Strangely enough, we don’t record fathers as a specific relationship between people – fathers are rather superfluous in births, so we don’t bother with them. We can carry out actions, and then we have the specific things to deal with – the transfer of title and custody – that we talked about earlier. I can *join a group*, I can *leave a group* (separated and added to a group), and then I can *dissolve a group*, *destroy an object*, *transform an object* and *die*. Obviously I participate in my own death – perhaps not today, though. Then we can modify things - I can *take things away*, *diminish things* or *augment* them. I can *add things* to something, I can *produce* something - make something - and <with> *brought into existence* I can *create a group.* The the birth of them brought into life: my birth brought me into life and was by my mother. I can create intellectual objects, and we have the special case of creating a type in my E55 typologies: this allows me to talk about who created the type, where was the type created if I’m making a thesaurus. So I can document the history of my thesaurus using this P135. So <from> the general case down to the very detailed cases. If I just want to know where somebody was involved, and I don't want to know in what particular thing - I don't want to know just about their birthday, I want to know *anything* that they were involved in, I could search for this and all of its sub-properties and get my death, my birth and who my mother was, and all of these different things, from just searching for the general case. Or I can go down and say: “I’m only interested in things where they actively did the thing, they performed it.” Or I can just say: “I’m just interested in things where they gave the title to another institution or another actor, perhaps to a museum. So I can search at different levels of granularity and I gain everything that’s below automatically, as part of the search mechanism. This gives me very great power for doing general searches, and then honing in to particular aspects of the behaviour of an individual or an object.

## Slide 23

As an example, I can look at *terminus post quem* and *terminus ante quem*. Here we have the famous meeting of Attila the Hun and Pope Leo I. Obviously this meeting happened after the birth of Leo I and of Attila: the birth of Leo I was before the meeting, and the birth of Attila was before the meeting. And the meeting happened before their deaths. Now Pope Leo I carried out this meeting, he participated in it, and he also participated in general in his birth. He was brought into life by his birth, and died as part of his death – it’s convenient if you die as part of your death. So if I know this date, then I know that this must have happened before it. But I don’t have to know precisely what the relationship is – I can take any *took out of existence*. So I can query for any object that participated in this meeting. I can look for the table as well as the people, and look for the destruction of the table as being something that happened after the meeting as well. By using the general case *took out of existence*, I can look for the dates when *anything* which was involved in this meeting was taken out of existence: the table, the people, the pen that they used – everything. I <can then> use all of them as limits on the dates of the actual meeting. This gives me a great deal of power without having to search every possible link between an object – in this case Pope Leo I – and its destruction. I can use the general case of *took out of existence* and find all things which were taken out of existence, which participated in this, and take the earliest one of those dates as being the latest date that the meeting could have happened. So if we had a meeting and immediately burnt the table because it was contaminated, and I knew when the table was burnt, I would know that the meeting happened before the table was burnt. And then I have my links, because these deaths, births and the meeting are all things that happened in time. They’re all events, each of them has a link through to time via a time-span and some dates we can associate with it. It’s a very powerful mechanism, and by using the general tree structure here, I get to do those queries at the highest level and get all of the specialisations below them automatically, because they're already defined for us in the standard. So we don't have to have very complex queries to find those specialised cases when we just want to use the general case.

## Slide 24

We can think of these events happening as world timelines, and we model all of these things as meetings, so all these events are the meetings of people and persistent items to give us a history. Here we’re going to investigate the life of Caesar. We have what we call a coherence volume - a block of space-time within which Caesar was born, and Caesar's mother was present at that, and Caesar started his existence, he was brought into existence by his birth. And then his life is this blue arrow through time and space, and he reaches the coherence volume of his death. Little did he know that he was reaching it, but he reached it. And he meets Brutus's dagger - in a rather brutal fashion hands - at he stops, that’s his death. The dagger continues on, because obviously the dagger was made before it stabbed Caesar, and it continued on after the stabbing. And then we also meet Brutus, so Brutus comes into this coherence volume; the dagger and Brutus and Caesar all meet; Brutus continues, Caesar doesn’t and the dagger continues. So we have this world line of the various objects. We’ve got a world line for Caesar's mother, one for Caesar, one for Brutus and one for the dagger. And then we have these volumes, which tells about the space-time within which these things met. So these boxes are the temporal entities in which these things met and that allows us to model historical events as though they were meetings between people and objects.

## Slide 25

We can also do this with things like depositional events, so in this case we're going to look at the eruption of the volcano on Santorini – Thera. We have some ancient Santorinian who comes along and builds a house, and we have a volcano which continues throughout. Then we have the volcanic eruption, which happens at a particular point in time, and the house stops being a house. The matter stays the same, so it’s *transformed* from a house into ruins, and we now have the lava and ruins instead of the house. This volume is the space-time in which the eruption happened, and we've converted a house into ruins, so we have a transformation here. We can look at things being deposited in an archaeological sense as a meeting between the thing that’s doing the depositing and the thing that’s being deposited.

## Slide 26

Another thing we can look at is not just the physical things we can model this way. We can also model information. Here we have the great Battle of Marathon, and the story which forms the reason for the modern running of the marathon. There were a bunch of soldiers who fought the Battle of Marathon and they won. This is the coherence volume for the Battle of Marathon and the famous victory, and then we have this runner, who sets off. He participates in the battle, and then sets off to take the information back to Athens. He reaches Athens, and the information object - the concept of the victory at this battle - is held in his mind and he gets to Athens, and he conveys that victory to an Athenian, to the first person he meets and then he dies. So his timeline stops, but the Athenian now carries the information of victory, because we’ve transferred that conceptual object from one carrier – the original runner - to the new carrier - the first Athenian he meets. And they go on and pass that information to another Athenian, so the next person learns it and we now have *two* copies of that conceptual object carried in *two* people's minds. Then it obviously escalates and everybody gets to know about it, right the way down to us knowing about it. Each of these meetings where we’ve transferred the information from one carrier to another – it could have been written down, of course, or we could have written it onto a CD-ROM or typed it into our computer, those are all <instances> where we are adding another carrier to that conceptual object – each of these meetings happens in this coherence volume and then the information object carries on, the conceptual object carries on. In these two cases the carrier - the person - carries on as well. But in this case we passed it on and the first character died. So now we can think of exchanges of information, the movement of conceptual objects, as being a series of meetings between carriers and the conceptual object. And we can pass them on by having an event or an activity which creates a new copy in some new carrier: writing it down or transferring it from mind to mind by telling. So we can use the same conceptual mechanism to deal with many different types of historical activity.

## Slide 27

How do we deal with time? Well, we use this general model that an event is *not* a point in time. There is no such thing as a point in time: something always has duration and so we consider it to be a curve, a probability curve of that event happening at this particular time. We can add some boundaries to this: if we know it happened “after here” and “before here”, so after is “this way” and before is “that way”, we know that is at “sometime within”. We call these the outer bounds. We have a property that allows us to say that at some time within, the thing that we're interested in happened, so it gives us the outer bounds of the possibilities of that thing happening. We might also know that it was definitely occurring at a particular point somewhere in there: the thing lasts for a week, and we know it was definitely on this day, but we don't know whether that was the first day of the week that it happened or the last day, so somewhere in these two weeks it happened, and it's a week long. That is the duration here, but it was definitely going on on Tuesday 1st. This is the way that we’re going to deal with it: we’ve got what we call the *inner bounds*, the *outer bounds* and the *duration*. By using those three mechanisms we have the ability to talk about events happening where we don't know precisely where they happened - we don't know perhaps quite what their duration is but we have a general idea of it. In addition to this we also have the traditional Allen operators, which allow us to say that something happened after something else, or all began at the same time as something else but finished before the other thing finished. All of the Allen operators for temporal relationships are also available within the standard. This allows us to relate to time primitives, but we can also do relative dating, so we can say: “This happened before that” as well as: “This happened sometime within these bounds, or definitely covered these inner bounds.” Inner and outer bounds plus relational algebra for temporal extents are all available within the standards, so we’ve got a very good ability to deal with the various types of dating we see from archaeological and historical sources.

## Slide 28

Let's now consider the Time-Spans property, and how it relates to other things. The time-span is itself a CRM entity: these double arrows show the isA relationship. A Temporal Entity is a CRM entity, the time-span is a CRM entity, and we also have the Time Primitives and the Persistent Item being CRM entities. A time-span can fall within itself or be contained by another time-span – that’s the part decomposition. The temporal entities have the time-span, and the time-spans relate to the time primitives ongoing throughout, which are the inner bounds, and At Some Time Within, which is the outer bounds. We can have names for the time-span – we can have time appellations, a special class of the E41 appellation we looked at earlier, applied to the time-span. And then we have a period, which is a particular type of temporal entity – an event or a type of period, a sub-class of period. And we can have the Falls Within and Forms Part Of breakdowns of period. And then we have Condition State as the other top-level sub-class of Temporal Entity. Period is bounded by place, Took Place At. These are all of the relationships we’ve looked at individually, but now in a graphical format that we can use to try and explain how they interact with each other, and give us some idea of the way different classes are related to other classes, and the properties that do the relationships: the isA property and the specific properties, the P properties that we have within the model.

## Slide 29

We’re also going to look at inheritance - the E7 activity we visited before - and we’re going to show how it inherits properties from further up the tree. An activity is an event, is a CRM entity, but it goes via temporal entities, so we use this dash line to show that there are some things in between. The E55 type is a CRM entity as well, but are some things in between. A CRM entity has the ability to have a note attached to it, and we have the general case of an E62 string, which is a type of primitive value. We generally think of an E62 string as being as a string of text – that’s what we’re used to in databases. But we have a more general concept of string within the CRM, so this could be a string of text, but it could also be a photograph or a video file or an audio file - any binary large object or ‘blob’ can considered to be a string. Everything that we have in the E1 CRM entities can have a note, and the note has a type and we can say that this is a public description of it / a museum description of it / an audio file of her showing what the bird sounds like / what the bell sounds like when it's rung, or whatever the type pointing to the string is. This means that because E1 CRM entity has this ability to have a note, that E7 activity - which is via event an E1 CRM entity - can also have a string. So I can have a description of that activity, or the sound of that activity, or a film of that activity attached to the activity via the P3 Has Note. CRM entity also has this P2 Has Type, which allows me to classify everything – this general case P2 Has Type is attached - so I can type my activity. “This type of activity is bell making” or “Running around uselessly – oh sorry, football”- we can attach any of these things here. In activity we also have the new property P14 Carried Out By. Remember we had General Participation P12 up at the event level, and then when we got down to activity we had the more specific Carried Out By property, so here is the property Carried Out By. I can point from the activity to an actor, an individual or a group of individuals who are going to perform this activity. I can then say in what role they were doing that activity, so we could have me as the presenter, or the photographer, or somebody who is paying for a church to be built, or the architect or the builder. All of these different roles in the activity can be differentiated by saying that the activity was carried out by somebody in the role of architect, in the role of photographer, in the role of sponsor and so on. There is another high-level thing attached to E1 CRM, another high-level property, which is P1 Is Known As. This allows me to link appellations – the general naming that we talked about when we looked at the big slide of the overarching view is also implemented as a property of E1 CRM entity, so that everything inherits it.

## Slide 30

Now were going to look at activities of measurement, which introduces the concept of shortcutting. Here is my measurement, which if you remember was a specific case under attribute assignment. An attribute assignment is a special class of activity, and I can assign the attribute to a particular thing. Any E1 CRM entity can have an attribute assigned to it, and assign the particular attribute. So here I am producing a triangle and the thing that I'm applying it to is an E1 CRM entity. In general cases I assign the attribute to it, but when I’m doing measurement I'm *measuring* that thing. I observe the dimension - so this observation, the dimension, is a sub-property of Assigned. Measured is a sub-property of Assigned Attribute To. And then the dimension itself has a measurement unit – centimetres, stone, colour, whatever it is – and a number associated with it. This was 6cm long, so the measurement unit would be centimetres and the number would be six. This is the shortcut: P43 Has Dimension, which allows me to say: “This thing has this dimension”, without producing the measuring event in between. This is the shortcut directly from the object to the dimension. I can optionally put in the measurement: it depends on when I'm creating my data from my original data structure. In my original data structure I’ve just got a field that says length, and in my desk instructions I’m told to fill that in in centimetres. So when I take that out and make everything explicit within the CRM, I say that the measurement unit is centimetres. That doesn't appear in the database, that only appears in the desk instructions that tell me what to put in that field, and the number that’s in the field. Then the dimension is Length Of This Object, but we don’t know what the measurement event is. There is no mention of who did this measurement in the original database, so I can just shortcut it to the thing. If, however, I think I'm going to integrate this with another datastream - say the lab notebook to the person who does all measurements - then I could create this measurement event ready for me to hook that other datastream into, because that reference from my workbooks would have the object number that is associated with the thing that that I did the measurement of. So I’d be able to link through and say that this measurement, the measurement of object X occurred, and then I’d be able to say who did it, when they did it, what the temperature was and the weather, because that would obviously affect the measurement. For example, if I’m measuring some paper, the paper changes dimensions depending on how humid it is, so I could say what date it was done on; under what conditions of humidity; what measurement technique I used; which instruments I used to measure it with. I can include all of that from my notebooks about the lab techniques that were used in this event. But if I don’t have that information and I'm never going have it, I don't need to bother creating the measurement event, I can just go directly from the thing to its dimension. So it gives me an option as to how I’m going to integrate data. mention they given options to how I am going to integrate data and which data I’m going to integrate along the way.

## Slide 31

That’s measurement - now we can go on to condition assessment, which is where I'm going to carry out the activity of assessing the condition of a particular thing. I can have the activity of condition assessment being the overarching activity, but it might be broken down into many subparts, and again I can use the ability to talk about different pieces of the activity. I can talk about the preparation for the activity; the first part of the measurement; the second part of the measurement; the third part of the measurement; the finalisation of it, by having different sub-activities within the condition assessment, and these could each have a different type, saying exactly which technique am using and so on. It <the measurement assessment> concerned this physical thing and measured this condition-state. Then I have a shortcut <called> Has Condition, so that if I don't know any information about how we assessed it / who assessed it / when they assessed it <and> in the database I just have the statement that it is “Good” condition, then I can just go straight from the Physical Thing to the Condition State and say Has Condition. This makes what we actually see in real life databases easier to translate into this, and it means I don't have to generate lots of condition assessment events which I’m never going to use to do anything with. It allows me to do simplification of my mapping.

## Slide 32

Now we’re going to talk about acquisition. This brings in a different type of shortcut – a specialised type of shortcut. Here we have the note capability and the type capability we’ve seen before. We have an event, and then we go down to an activity, and now we’re talking about the acquisition. Here we have *two*  links, two properties, from acquisition to actor. One is who transferred the title - Transferred Title From - and then who the title was transferred to, so there are two different instances of actor, one of which gave up the title and one of which received the title. This is the reason we have *two* links between acquisition and actor, showing us different things. The acquisition transferred the title OF this physical thing. These two shortcuts are special instances, and really illustrate the difference between different desk instructions producing different mappings within the CRM. If I have a field which says that somebody owned an object - they have ownership of the object - and I don't maintain that field, I fill it in when I first get the object, but I don't ever bother to update it, then I have the case that I have Former or Current Owner, so the actor is the former or current owner of the thing. But if I religiously maintain that field, so that every time the ownership changes I go back into the database and I update, it then I have a much stronger statement to make: Is the Current Owner Of. So the difference between these two fields isn't the relationship between the field Owner and the object – the thing that the record in my database is about – but it’s actually the way I maintain that field that’s important. If I keep it up to date I can make the P52 statement, but if I don’t keep it up to date or I *don't know* whether I keep it up to date, then I have to make the weaker P51 statement. These are trying to encapsulate not just the data structure and what goes into the fields, but also how we maintain the data. So when we encapsulate the data in the CRM, we're actually encapsulating not just the data and the thesauri and wordlists we use, but also *how* we maintain that data. It's all being included, so that when we ship the data out to somebody else, they'll understand what it is that we're saying. They have all the semantics of the database - they get the desk instructions, the thesauri *and* the data all in the same structure, so they don’t have to go and look something up to find out what it is - it's in the data that they get. That’s quite an important extra functionality we have in some of the shortcuts. Now obviously this doesn't matter if I have a series of acquisition events. If every acquisition is documented and I know who gave up the title and who gained the title through the acquisition events, and we have the time that these happened, then I’m able to show a complete history of the ownership of the object. It’s only when we *don't* have this information and we only have *this* information that we can differentiate, or we *need* to differentiate between the two different ways of looking at the data <and> how we maintain it.

## Slide 33

Moving – here’s my physical thing. It has a former or current location, which is the weak statement, and a *current* location if I’m constantly maintaining the information. <Then> we have Moves, which will move something from somewhere to somewhere, and the thing that we're moving obviously. And then there’s this extra thing – an extra shortcut which really labels this as a museum standard. It’s one of the few things that is museum-only: the concept of Has Current Permanent Location. The current permanent location is where the object should be if it’s not somewhere else. The idea is that we will have a place in the store, in the basement of the museum, where the object would be put if it wasn’t on display. In some museums, the object could never be in that current permanent location, because they use it as a way of doing their inventory control. So you’ll get shelf marks for a little bit of shelf – a two-inch length of shelf – in which a 400 tonne steam engine is supposed to fit. And that’s the current permanent location, so the object may never have been in its current permanent location, it’s just where it should be in theory if we taking a stock control of it. That marks this as a museum standard; it's one of those things that *only* museums are interested in. But we can ignore it if we don't need it for what we’re doing. So if we’re trying to use the idea of a move outside a museum - within an archive, say, or if we’re looking at a library or if we're talking about a medical database - then we just don't need to use that property, because it’s optional.

## Slide 34

This is the general map of Making Things, so we have Modifications, and the special case of that – Production, where we’re going to make something. If we’ve modified it or made it, it is necessarily man-made. We can use designs or procedures in order to say how we’re going to do it, how we going to make it. Designs or Procedures were originally only for Modification and Production, but we’ve now realised that we can use designs or procedures for any type of activity, so it's been generalised to give us the ability to say: “I measured this using this procedure, I acquired it using this procedure, I assigned an attribute to it using this procedure.” So we can do all sorts of things with Designs or Procedures.

Designs or Procedures *might* use a material as part of doing the Design or Procedure, even though that material doesn’t end up in the object. If, for example, you’re doing lost wax casting, where you make a wax model and then put clay around it to form a mould, and then you melt the wax out so that you can put the metal in to actually create the casting, then there is no wax actually in the final product, because we’ve taken it all away to create the void that we’re going to actually put the metal into to make the casting. So we can talk about a Design or Procedure using material, even if that material doesn't end up in the final object. This gives us a great deal of sophistication for the way we model production techniques and map modification techniques. Obviously, modification is an activity, so we can talk about an actor carrying out the modification or production in a particular role: we can talk about the artist, the caster, the engraver <or> the various different roles there are in producing or modifying something. We can have a whole variety of different roles, and we store all of those different roles in the E55 Type class, so that we have a set of classes <or> types of things within the E55 Type which relate to the roles that people can perform particular activities in.

## Slide 35

Now when we we’re actually producing the standard, we have two ways of showing you what properties a particular class can have. In the published standard - the ISO document - we only show the things that are declared for a particular class, and these are the things that are in bold here. Here’s the declared property of E11 Modification: P31 Was Modified By, a physical man-made thing. This modification acted upon this thing, this physical man-made thing, which, if we look back to the previous diagram, is this relationship here: P31 Has Modified. That’s declared as part of the definition of E11 Modification, but remember that Modification is an Activity > is an Event > is a Period > is a Temporal Entity > is an E1 CRM Entity, so it gains all of the things that it has inherited from there. Here is P1 Is Identified By, so I can give it a name; P2 Has Type, so I can classify it; P11 Had Participant, as opposed to someone who did it, which was down at the activity level - P14 Carried Out By. Occurred In the Presence Of; P16 Used Specific Object, so I can say that I used this particular mould for doing my lost wax casting, for instance, and in what mode. All of these *other* things I’ve inherited from further up the tree. We don’t show those in the very small ISO document, but we do have a different version of the document where all of the inherited properties are listed. That's available as a text document or it's available as an HTML document, where you can zip around inside it. Just by clicking on P12 Occurred In the Presence Of, it will take you to the definition of P12, so you can just check the scope note to make sure that you understand it - you can whistle about in it. All of these inherited properties we list in *italics*, so that you understand they came from higher up the tree. But the ones that are actually declared for this class are in **bold**. In this list we’ve got inherited properties; declared properties; some more inherited properties; another couple of declareds; some more inherited; and another declared. E11 Modification in the ISO document will just have the four statements underneath it; within the more useful, more friendly document with all of these things in - in the HTML version - you get all of these, and all of the classes and property names are hot-linked through to the definition of that class or property, so it’s very quick to move around. This is very useful when you’re first using the standard, as you're getting used to things, because you don’t have to hold all 137 properties and 86 classes in your head before you can use it. You can use this document and whip around inside it, to very quickly find your way around. Then over time you'll have to do it less and less, as you get used to where things actually are.

## Slide 36

This is a general case for the way we can change things. This is talking about modification, where we can do part addition: I can add a piece to it, add an extra leg to my table to make it into a five-legged table. Or I can do a part removal: I can take the fifth leg away and make it back into a four-legged table again. I can transform things, where I’m changing the identity but maintaining the material, as opposed to a modification where I’m changing the material but *not* changing the identity. These are all happening to persistent items - things I can look back at and know are there - physical, man-made things we’ve had some kind of human interaction with, that we’ve modified. So that tells us about how we modify things or transform.

## Slide 37

And here we have another case, where we're talking about taxonomic discourse. We’re talking about how we create the typologies that we use to classify things. The concept of a type is a conceptual object - it's a product of our mind. So we have the activity in general; the creation event which allows us to create the conceptual object – that’s where we think of something - and then we have the special case of that, which is the type creation, where we’re creating a type, generating the concept of a particular E55 type. We can create type in that - here’s my E55 type which was created by it. But I can *use* real world objects in order to help me think of that. Particularly, say, in Natural History, I’ll have a particular example of “the bumble bee” or “the insect” that I’m going to create the type for. So I will say: “Was based on a particular instance of this insect.” And I can say what *role* that object was used in doing that, so I have this taxonomic role; I can say it was a lectotype. This particular bug was the lectotype for creating this type. Here’s my type creation, I used this particular bug as a lectotype to create this type. I can then point back at the thing and say: “This is the one that was used in the role of lectotype to allow me to create this,” so I've got full control of exactly how I created the type and which objects were used to help me create the type. I have full documentation of the way my thesauri were built up and how I created this stuff. So when I export it and I want to look at that date again, in 15 years time or 30 years time or 150 years time, I know exactly how the typologies that were used to classify the data were created. They may have long since been superseded, and if I want to use that classification in my research now, I need to understand how that classification happened. This allows me to document the metadata that goes around it - the procedures and all the classification systems, completely and thoroughly – and incorporate it into my database, export it. So I’ve got everything about how I used it. And then of course I can then actually classify something with the type using my type assignment, so I’m actually saying: “Right, this one … this is a clicker, this thing in my hand is a clicker. But I can also talk about *how* I developed the concept of a “clicker”, and which object I used as my type specimen for saying: “This is what clickers are.”

## Slide 38

We’ve talked a lot about physical and immaterial things, material and immaterial things, so it’s a good idea if we actually look at the structure of what these things are. This is the tree structure showing the hierarchy of physical and conceptual objects that we looked at at the very top level in that original diagram. Here’s a thing – we can have legal objects, that is objects that can have rights on them - those are things like ideas which can be reused - identifiable reusable physical things and conceptual objects. We can have man-made things: obviously all conceptual objects, all the products of the human mind are man-made, but some physical objects are not man-made, because we could be documenting natural materials from a natural history museum, for instance. We could be documenting a rock sample, as an example. We have physical things, and then we get physical man-made things. We also differentiate between physical objects - these are things which have real boundaries. This clicker is a physical object, whereas the writing on the side of it is a physical feature, in that there’s no real way I can pick it up and transfer it around. So things which are generically holes or scratches we would consider to be features: a canal is a feature, a cave is a feature, but this clicker and a table are objects, and they can be both natural. A cave is a natural feature, a boulder is a natural object. We can have physical man-made things, and they divide into man-made features and man-made objects, so that we have objects which have been made by somebody and features that have been made by somebody: if I cut something into a rock, that’s a man-made feature. We have biological objects - a special case is of course me as a person – I’m a biological object. A bumblebee, an insect, a lion, the tiger that we had earlier, and a snake are all biological objects. Then we have the special case of a site, and that's where we've taken a physical feature and we’ve just drawn a line around something and said: “Right, this is the site.” So any kind of arbitrary distinction, so-called fiat object in space, is a site. Now you remember we talked about Place being the mathematical construct on the surface of some larger constellation of matter: the inside of my wedding ring; the back of the coin; or the Earth. Here is the difference: the site I can take a photograph of, because it’s the particular stuff that’s at a particular site at a point in time. So I can photograph sites, but I can’t photograph places. Many people find that a useful way of distinguishing between a place and a site. Site is the stuff that’s at a place at a point in time, so I can photograph it, whereas the place itself is just a mathematical shape; it’s just a definition of where stuff is, I can't take a photograph of that. Collection is a set of man-made things which were going to curate in some way: a collection in the way museums talk about collections, or archives talk about collections. Another special case of man-made object is an information carrier. Information carriers are things that are designed to carry information. They don't have to have any information on them, but they’re designed to carry information: blank CDs, blank floppy disks, blank notebooks are all information carriers, whether they’ve had anything written onto them or not. However, if I gouge my name into a table or into a prison wall, that doesn't make the prison wall or the table an information carrier, because it was not intended to carry information. The fact it now does have my poem inscribed on it is irrelevant; it's not an information carrier. I can still represent the fact that it's carrying a copy of my poem - the properties are there - but it's not *designed* to carry information, so consequently we don't consider it to be an information carrier. Then we can go down to the immaterial things: I’ve got this nice red line to show that this is *material*, so this is things I can smash windows with, and this is *immaterial*, where if you know it I’ve got to shoot you – the immaterial world. We’ve got conceptual objects in general, and we break those down into propositional objects, things which *say* something about something, and symbolic objects, which are just the image. Normally, in a book we’ll have a symbolic object which *carries* the propositional object. We’ll have the *text* as it’s laid out, with a particular typeface, a particular set of spacing and a particular font - which is the symbolic object - and then the *meaning* is the propositional object that’s carried by that set of symbols. Because I could change the type face and it would still be the same propositional object, but it would be a different symbolic object. That's what these two allow us to talk about. Appellations are symbolic: they’re names for things, they don’t carry a proposition. They don’t say something about it - they’re just a name for it, a reference to it. If you think of the artist formally known as Prince, who changed his name to a symbol, that’s an ideal understanding of what's going on. His appellation - his name - was now this symbol, not a word, not like Stephen. That's the difference between an appellation as a symbolic object and this idea of carrying information - that there’s a proposition in there. There is no proposition in the symbol; it's just his name as represented by that symbol. Information objects are a combination of a propositional object and a symbolic object, in that they're carrying information in a particular form on to the outside world. Information objects – the stuff that’s in the book – is a combination of the typeface and the layout *and* the thing that’s being said in those words, in those pictures or diagrams. An information object is a combination of both symbolic and propositional objects. And then we have these various classifications, these various sub-classes of it. An E31 document is *not* a novel. An E31 document is a very technical use of the word document. What we mean by an E31 document is something that deliberately sets out to document something in the formal sense of documentation. So it might be that it isn't text: if I take a photograph of a museum object with the intention that this will document the museum object – so that I’ll know what it is and what state it's in, and I can re-identify it - that’s an E31 document, even though there’s no text in it at all. I could have a sound recording of a bell, which is intended to document the sound of that bell ringing, and that would be an E31 document, even though it has no text and it's only playable as a sound file. So E31 documents intend to document something. And then we have a special case of that - an authority document, which documents our types. So this is where we can create our thesauri, which tell us what our different types are. This is the authority document that documents the types in a formal sense. Then we have the Designs or Procedures - how we do something - and we have general case of Linguistic Objects, which are our texts and so on. <Then> we have special cases of Inscriptions, and a special case of Title. Title is the name of a book or a painting or something like that, in the formal sense that it's a name, but it's also a Linguistic Object. Really, what we're saying here is that the title can be translated. If the appellation of a thing can't be translated, <if> you can't say *The Blackbird*, or whatever *The Blackbird* is in French - it’s gone straight out of my mind – then it’s not a title, it’s just an appellation. “Stephen” has no translation, so it's just an appellation. But if I have the name of a book which I can translate into another language then it's a title in the formal sense of a title. I also have Visual Items down here. Visual items can be images or they can be marks, so there we’re talking about trademarks and little symbols, like the red triangle for Bass Beer and so on. They might be inscriptions, they might be scratched into something. All of these are immaterial ideas. We’ve talked about type a lot – type is a conceptual object, and we have special cases of type: Measurement Units, so we can talk about centimetres or furlongs or pounds or kilograms. Material: things that we used to make stuff, but we’re not interested in documenting the individual thing. And this produces an interesting choice of scale. If I am documenting building materials, I'm probably not going to document every kilo of sand that was used to make up a building; I’m probably not interested in which kilo of sand came from which bit of which desert to make the cathedral. I’m probably not interested in each individual brick that makes up a fairly standard 1930’s townhouse, and I'm not going to record the manufacturing process of each individual brick. They would be materials, because I’m not interested in documenting the individual pieces, I just want to document the fact that this type of material was used. I want to document that bricks were used, that sand was used, that thatching was used – reeds to make the thatch on the roof. But if I'm interested in Roman bricks, each individual brick is of vital importance to me, because I’m going to record: how it was made; which type of stamp it has on it; which techniques we used; which clay was used, from where, in order that I can actually produce a really detailed understanding of the way Romans made bricks. So it depends on my research question or questions at what level I draw the line between something being a material and something being worthy of documentation. One person’s database might document things that another person's database would just consider material, and the nice thing about the CRM is that we don't care. We don’t tell you where to draw the line; we can just absorb your data no matter how you’ve done it. So if you consider gold to just be a material that's fine, but if you're interested in tracking every individual gold nugget that came out of a particular panning operation or mining operation, we can do that as well. We don't care what your research question is, we just allow you to include the data that you've generated as part of documenting your research question into the general milieu of information across the whole of the cultural heritage domain. I’m going to deal with appellations in a couple of slides, so I’m going to leave that bit out, because I’m going to go into that in a little bit more detail in a few minutes. But that’s covered all the different pieces of the physical world and immaterial world, except for appellations.

## Slide 39

Here’s a little bit more detail about the Actors. The actor is somebody or a group of somebodies who have responsibility for their actions, and they’re divided into E21 Person – remember we just saw that E21 Person was a subclass of Biological Object, so as a person you are both a biological object and an actor.<As for> the groups, we also break down groups of individuals into legal bodies. Legal bodies are groups which have some legal identity. The FORTH Institute is a legal body, the UN is a legal body, but a group of people acting as a collective might not have legal status.

## Slide 40

E53 Places we’ve talked about a few times now. A place is an extent in space, so it’s just the mathematical limit. It’s diachronic, and it doesn't matter when that mathematical limit is true all the time. It’s always relative to some larger persistent constellation of matter, so the persistent larger constellation of matter might be my wedding ring. The inside of my wedding ring is a place, but it doesn't have a latitude and longitude; I can't give a UTM coordinate for where my ring is, and in the course of this lecture I've moved around lots and lots of times. But the mark on the inside of the ring which tells me who made it and what type of gold it's made of is always on the inside of my wedding ring, no matter where my wedding ring is. So it gives us the ability to reason about where something is even if we don't have a grid co-ordinate, as many databases insist on. So we can talk about things on the bow of a ship; an African dinosaur footprint which is currently in Portugal, because when the dinosaur made the footprint it was in Africa, and that piece of Africa is now Portugal in Europe, so <though> it’s moved it’s still an African footprint; and where Nelson died, which is inside the *Victory*. I have no idea where Nelson died as a lat<itude>-long<itude>, because the ship was sailing around with him in it, but I know precisely where he died because there’s a plaque there and I know precisely where it died givers a plaque there that says: “He died here” and I can go and point to it. That’s now in Portsmouth, in the dock in Portsmouth. Well obviously he didn't die there, he died somewhere in the Bay of Biscay, but I know where it was. So that gives me this idea of place; I can point to the place relative to some larger constellation mapped. Now of course many of these, when we’re talking about archaeological sites or architectural buildings, are to do with a continent. We're saying: “It’s on this continent.” But we can also talk about the front of a coin, or the the top left-hand corner of a painting, or the bottom right hand corner, where somebody’s signature is. So it doesn't matter where I hang the painting, the signature is always in the bottom right-hand corner of it. That’s what place does for me.

## Slide 41

The properties of place give me all sorts of capabilities. I have the ability to name it: now that name could be an address, it could be a place name, it could be the spatial co-ordinates to it, or it could be a section definition, like the inside of my wedding ring. So here's the physical thing, and this section definition says it's the inside of this physical thing - my wedding ring - and that’s the place appellation which applies to the place where I'm going to say that this, the manufacturer’s mark, is. Obviously periods take place in a place. The period is then something happening, and that can take place at a place *on* a physical object. So we can talk about the *making* of that mark on the inside of my wedding ring. That is a modification event, which is a subclass of period, so it comes between production and period in this tree here. Here is the former or current location which allows me to talk about a not well-maintained database, saying where something is. I also have this section definition - Has Section - so I can shortcut going through the appellation route to get that physical thing, to say where it is. So I can talk about the inside of my wedding ring without going through the long route – I can use this shortcut to get to that.

## Slide 42

I said I was going to revisit appellations. Remember, these are symbolic conceptual objects. Here's my appellation – that’s the general case of appellation - and I have specific appellations for things which I know are actors. I can talk about actors’ names: they can be Stephen or FORTH…FORTH an actor appellation, the name of this institution. <Then there are> Identifiers, which are the identifiers for objects and things that we’re interested in studying. We can have rules for how identifiers are made up, and they’re covered in an extension to the CRM called Functional Requirement For Bibliographic References, Object Oriented – FRBRoo – that’s an extension that we’ve done already. Identifier allows us to talk about the strings of letters and numbers we used to identify things. We have Time Appellations, the names of periods of time, and the special case of that - the date. We recommend that you use the ISO standard for dates, where you put the year at the beginning, then the month and then the day, although you can use any layout of date that you want. It’s just that that is convenient because it sorts properly. You’ll notice when we created that example of that Yalta Agreement I was using, 1945-02-11, to make it sort properly. Conceptual Object Appellation, so if we’ve got a name for a conceptual object we can put it there. And then the special case of title, which - remember - are an appellation and a linguistic object, so we can translate them. Place Appellations – we just looked at those – place names, special coordinates, section definitions and addresses. And then we have this special idea of a contact point, so we can talk about *how* we gain access to an actor - how do I communicate with the actor? Well, that’s a Contact Point, and a special case of contact point is an address. So an address is both a place appellation and the name of a place that I can send information to. So if I want to post somebody a letter, then the address would be a contact point as well.

## Slide 43

Just to illustrate how extensions can work, this is an example of how we might extend the CRM to process some of the additional information that’s in the Getty TGN. The Getty records *more* information about geographic names than we do in the CRM, because that's not our research question. The TGN is the Getty’s research question about what places are called. We can add some extra things in, to help carry all of the data that is in Getty if we want to. We don’t have to, but it’s possible. So we can have this additional class which is place naming, where we’re talking about *who* named something, a particular thing. That’s obviously a subclass of attribute assignment. I can create a *new* extended subclass called place naming, and I link it into the existing CRM structure as an attribute assignment. Now, when I search for attribute assignment, I will get back all things which have been made into place naming, in a special extension database that’s been created. Then I can have some extra properties which link place naming to something. I can say that it assigns a name – I’ve got a place appellation assigned, and that’s a property. I can assign it to a particular community, and this place appellation is identified by a community. This community decides that this place appellation applies to a particular place, so that the community understands this place naming. I can say which group carried it out - it might be an official body that does it, or it might just be the people who live in a particular city <and> say that the city is called “this”. And it’s applied to a particular place, because obviously it’s a place name, it's applied to some place. So that’s the generic way of extending it.

## Slide 44

That’s the generic way of extending it, and then as an example of that extension we can talk about Nineveh, which has been known as two different names - Nineveh and Kuyunijk. The community of people who live in the city of Nineveh - the cultural, social phenomena – the people who live there over time, have done this naming. The group of people is the people of Iraq, <they> carried out this naming to this place, and here we’re referring to the TGN of it, to point to that. They've assigned this name at one particular point in time, and they’ve assigned *this* <other> name at a particular point in time. I can identify when that naming took place by attaching time-spans to this. Here we say that this one was applied in the 20th century, and this one applied at a different time arms. This social cultural phenomenon - the people who lived in that city – have a time-span of the first millennium BC. So I can say that this community identified Nineveh as Nineveh, but later on a different community has identified it as Kuyunijk, and now it has gone back to being called Nineveh. So I can model those different changes in the way people name things just by using this extension, but the extension is fully compatible with the original. So if I have data that uses the extension I can still search it using the mechanisms that are available to me for searching general cases of CRM data.

## Slide 45

Let's move on and think about visual content and subject. This allows us to talk about how we can depict something on something else. Now remember we talked about Information Object earlier, and we said an information object is something designed to carry some information, but we can also have other things carry information. We have physical man-made things, of which a special case is Information Carrier, and they information carrier is designed to carry something, but in general the desk that I’ve carved my poem into, or the T-shirt that I’ve got a picture printed on (maybe that is designed as an Information Carrier, who can say?) depicts, so this object has on it an image, it depicts some CRM entity. So we have the Information Object, which is being carried by this and refers to this: it could be a poem or could be a picture, whatever it is – if it’s a picture, it’ll be a visual item - and we have the special case of showing a visual item representing a particular thing. We also have the Mode of Depiction, which allows us to talk about a planned view of something, or an elevation of something: a particular way of showing an object as opposed to a style, I suppose. And then the Visual Image is where we’ve got an image which is a visual item, because visual item includes things like trademarks as well. So this is Images, this is more general case of any visual thing, including marks and trademarks and so on. This allows us to show the way we represent subject matter on anything, whether it's a general physical man-made thing or the special case of an information carrier.

## Slide 46

That’s really covered the standard, and the way we can do things with the standard, so now we need to move on to thinking about some of the applications of it. The first way we can use it is obviously to take the content of different data streams, different databases, research databases, and represent them so that we’ve made them more general, so that we can get at all of the implicit things that are in the data structure in the way we fill it in - in the desk instructions, the thesauri and so on. But when we do this kind of mapping exercise, so we understand what's in the original database and how we’re going to represent that on the CRM, we quite often find that a field in the original data structure has multiple meanings, depending on what data is actually in there. So we actually have to produce mappings which say if this field contains this piece of information, map *this* way; but if it contains a different piece of information, then map *this* way. Just to illustrate that, I’m going to show you how to do mapping of Dublin Core data to the CRM. I’m going to do that by showing you two example mappings: one for a piece of text and one for a painting. Here’s a piece of text, which just to be completely cyclic of course, is the text about how to do this process, about how to map Dublin Core metadata to the CRM. This was a report written by Martin Doerr: here’s the front cover and the front page of that report. It was published by this institution, ICS-FORTH, in their technical report series. It’s Technical Report number 274 published in July 2000, and it’s written in English.

## Slide 47

So how does that map to the CRM? Well, the report itself is a linguistic object, because obviously we can translate it. It has a name - the mapping of the Dublin core metadata element set to the CIDOC CRM - and it was created in a creation event by an actor whose name is Martin Doerr. Remember <that> we're going to differentiate between the person and their name, because Martin could be known as many things: he could be names Martin, or Martin Doerr, or M. Doerr, or Martin is originally from Germany, so if you have is spelling of his name correctly in German, it has an umlaut over the “o”. But he now lives not in Germany, and people don’t have umlauts in their typefaces everywhere, so consequently he has discarded the umlaut for ease when he’s publishing. All of those are different versions of his name, but it doesn’t change who he is. So we have this actor identity, and we’ve just used a random number to generate the fact that he is an actor. He is actor number one in this example. It was used for a publication activity which was done by another actor, which is ICS-FORTH. So this is the name of ICS-FORTH – obviously there are many ways of writing ICS-FORTH, in Greek, in English and so on. So those are all different versions of the actor appellation, but it doesn’t change the fact that it's this legal body act of specialisation of actor that has done it. We’ve given ICS-FORTH actor number 2. It’s identified by a conceptual object appellation: FORTH-ICS Technical Report 274, July 2000, which from external knowledge we can say is a fourth identifier, and we can use programs which understand the format of names in order to say: “Oh, that’s that type of name..”, if we have that kind of structure within it. And it's written in the language English, so this is a mapping of that Dublin Core record, making explicit all the things which are actually hidden in the original records. Back here we’ve got this idea of creator and publisher, and inside there are hidden creation events. We’ve just got a string, which is their name, but we know that they might be known by other names - other strings – so we’re going to insert proper actors there. First their identity, to which we’re going to apply the appellation, which is in the Dublin Core record. So that’s pretty straightforward – the way we do that.

## Slide 48

Let’s go and look at a painting and see how that differs. Here’s a Dublin Core record about this painting: it's of Type Image, Type Painting, two different typologies in use within Dublin Core, two different classification systems. It’s called *The Garden of Paradise* and it was created – well, we don’t know who created it, all we know is that they created it. So they are known as the master of the *Paradise Garden*. We have a name for them, we know who they are, but we have no idea who it really is. And it was published by the Staedelsche Kunstinstitut. Publishing a painting is very different to publishing an article or a technical report. What this means is that this institution was-were the first people to document it in a formal way and present that document to the outside world. They were the first documenters of this painting.

## Slide 49

So how does that translate into CRM? Well, now we’ve got an Information Object, it’s not a Linguistic Object, because it's not a text. So when we look at that field which says that it's an image instead of a text, we have to produce a different mapping. Here it’s an information object, and this happens to be the information object’s real number in the museum of this painting. Obviously that’s not in the DC record, and we just happened to use that because we know it. It’s got the appellation *The Garden of Paradise*. It was produced in a production event, which was carried out by an actor. We don't know the name of the actor, we just have the mater of the *Paradise Garden*. However, we do have a very precise identity for it, because the united list of artist names knows <or> has an identifier for the master of the *Paradise Garden*, even though we don’t know who it is. It’s a very precise identity for someone we don't know who it is. ULAN is another product of Getty, like the TGM, and this is the ULAN number for the master of the *Paradise Garden*. Here we go – that’s his unique identifier. Remember that when we were doing the technical report we just generated some random numbers. Here we have a third-party number we can use, which is a solid number – a good URI for the identity of that actor. In the first one, when we published it what we were talking about was actually publishing the document, but here publishing is about actually documenting the thing. It’s documented in a document which was created in a creation event by an actor who has an actor appellation - the Staedelsche Kunstinstitut. We say that this publication was a publication creation, meaning that’s the type of creation that it was. And the information object has an E55 type of DCT1 image. Because it’s a DCT1 image, I can use external knowledge to process that, and say that it’s an AAT painting as well. By understanding this, I can add knowledge to it by merging it with another typology, another datastream. So at this produces a very different mapping to the mapping we had for text. By looking at a particular field’s value, I can determine that I need to do different mappings and different transforms of the data in order to produce valid CRM data that contains all the information that’s in the original datastream. So the mapping process not only needs to look at the data, but it also needs to look to see if different data values in the data produce different mappings for different types of object. This is one of the weaknesses of Dublin Core; it doesn't really show us all of that semantic richness in there, because it's only got a few fields. It's alright for its purpose, which was to do resource discovery, but it doesn't allow us to blend different data streams because the field used in different ways for different things. We don't have that same level of capability that we would do with the full semantic mapping we get with the CRM. But because the CRM is able to extract that information - all those implicit events and knowledge about how things are done - we can produce mappings which will take low specification data and actually enrich it into a semantically richer environment. So we can blend it with other data streams and mix it and still use it, even though it's not at the same level of granularity as the data perhaps from a very detailed art historians’ database about that particular painting, which is discussing the iconography of it, the techniques used on it, the different types of paint and so on. We could have very great detail in one place and much thinner detail in another, and be able to blend those different data streams seamlessly using the CRM to blend them.

## Slide 50

What lessons have we learned now from doing these kinds of mapping operations? The interoperability between different data sources can be defined by our ability to map the data structure. Data that’s got very poor or very thin structure is less semantically interoperable than data which has very rich structure. If you just get a piece of text about something with no structure in the text, we have very little ability to produce semantic interoperability with just that text, whereas if the data is all broken out of that text then we get much richer semantic interoperability, because the semantics from that free text have been extracted and made available to us to process. The mapping capability of a particular data source will show us how semantically interoperable that data source is. So if you use the CRM to help you build your data structures and you take a great deal of the CRM into your data structure, you know that you will have semantically rich data to pass to other people - you'll be semantically rich in your interoperability capabilities. The mapping of these types of networks is relatively simple. The databases are usually fairly flat, not very complex, and we reduce complexity by simplifying the data structures. So we have a field for artist, rather than having a field for the creation and who has created it and in what role; but we can extract very simply from that into the richer CRM context, because the rules for how that field is filled in are in the desk instructions. By using desk instructions we are simplifying our database structures to make it faster for people to work with. As long as those desk instructions are explicit, we can disambiguate the data and extract all those things which are implicit in the desk instructions out and make them explicit in our CRM data very easily, as long as we’ve got a copy of the desk instructions, as the Southampton story told us. It’s very important to embed your desk instructions and your thesauri into the data stream, so that when we export it we get everything: how you did it, as well as what you've done, <i.e.> the results of your research. We quite frequently find that we've got composite paths, so that for instance the artist and the date field for a sculpture both have part of the same path; so there’s the relationship between the object and its creation or its production and then somebody doing that production in the role of artist, and the time that they did it was the date. So the first part of the path is the same for both endpoints, and then we split out to do the different final pieces. And we find that that’s very common – that in the data structures we’re producing for our actual research work we’re doing these compressions, and we know we’ve done them, they’re implicit in our design, in our desk instructions. So we don't need to have all of these fiddly bits, because we know what’s going on. But when we want to make that shareable, we have to make those things explicit, and those composite paths are very common. That’s not something to be scared of, that's what we should be doing and that's the way we should design things, as long as we *know* that that composite is there and we can extract it on the fly as part of our export to CRM data. Along the way we’re going to make these nodes. Remember that we had the creation node and the actor node that we put into the DC records. We know we are going to create those, and so we just need to understand that we can create them as part of the process of building our original data source. If our original data structure mashes things together so that we can't produce individual nodes in these chains, then the data structure is not going to have high semantic interoperability. So we want to make sure that when we extract the data into the CRM format, we can identify those intermediary nodes uniquely and easily – they’re not two things crammed together. That makes things very easy, and then the cardinality constraints must not be enforced in the CRM and in the export. We may have / a person will have one father, but we might have knowledge about no fathers, one father or many opinions about who the father of somebody is, from different data sources. So we must not enforce those cardinality constraints to give us maximum interoperability; if we have many different opinions about who the father is, then we can carry them all in the CRM, we don't say: “There has to be *one* truth.” We give all opinions and then, as a scholar, I can make a decision about what's the most likely, <or> for all these opinions what's the most useful. We’ve also learnt from this <that> domain experts, cultural heritage people, actually learn this schema mapping very quickly. I run a tutorial for people to learn this: including doing this piece of the tutorial - what you're watching on video now – *and* going through and getting people to usefully do mapping, is two days’ work. Half a day to do this, and then within a day and a half people are actually able to do mapping of their own datasets very usefully. However, if you give it to IT experts they don't really understand the desk instructions and the processes which are part of our cultural heritage internal culture: the way we do things. We all have shared frames of reference which IT experts don't have. So they find it actually quite difficult to do the mapping without having a domain expert as part of the team. In fact, <it’s> probably virtually impossible for them. They either completely underestimate it <and> they think it's trivial - and it isn't - or they get bored by it because they find it too detailed, and they spent too much time trying to find out meaning and it just swamps them, because they don't have that shared frame of reference that we all do as cultural historians. So that can mean that the IT expert should be at the periphery of this process, and that the actual driving force in doing the modelling are the domain experts, the people who actually know the material. They can rely on the IT experts to help them out with how they do stuff, but it’s them actually understanding their data, and the *way* they create their data that’s the most important thing as part of the mapping process. We do need more intuitive tools for helping domain experts do this, and this is one of the open research questions within the CRM domain: how do we make tools which are helpful and supportive of this process? How do we bring together different identifiers? There’s a working group looking at how to deal with the co-referencing problem which we <saw when we> looked at back at the Yalta agreement. That’s one of the problems we identified there. And how do we deal with different technologies and reference them to each other? If one group of people call this a “castle” and another group of people have called it a “fortified place”, how do I make sure that we understand that they're actually talking about the same thing, that they're just using different terminology for it? So we need to go through that process as well.

## Slide 51

Throughout its development history, the CRM has met other ontologies. We’ve found that we’re a little bit different from many of them. Many of these different other ontologies lack an empirical base. One of the nice things about the CRM is that it has not been built from: “I think therefore I am”, down to deducing the existence of fairy cakes. It’s all worked from: “These are the data structures that we actually use in the discipline, let's extract from those upwards and find the things that are common across all these different sub-disciplines, different data structures that we actually use, so that we are actually grounded in real practice in our domain, in our cultural heritage domain.” Some of these others don't have that, they're just thought of from very high-level concepts and drilled down towards concrete things. Along the way it's very easy to get lost, whereas working from the conceptualisations that we use generally, we’ve got a very firm base. They tend not to have sufficient richness in the relationships in order to get the functions that we require out of it; they tend to be processed by looking at different terminologies and just processing the terminologies, so they are very good at sub-classing things, but they’re very poor at how things relate to each other, because those are not things which are encapsulated in the terminology. And they don't have any idea of what they trying to achieve, whereas we do; we’ve got a very strong idea of what we want to achieve, and it has been driven from the types of things that we want to do within the cultural heritage sector. So we have a good specification of what we are trying to achieve - interoperability in this domain - we have a rich set of relationships, and we’ve built from a strong empirical base. It does mean that we miss things out which we don't document: museums don't tend to document things like contracts; that's not actually part of the way museums work, and archives don't document contracts per se. They document the piece of paper that’s the contract, but not the content of the contract itself. So we don't have concepts that are not in the databases from our domain. But it detects content that isn’t made into lexicons, into thesauri. So we have this idea of Persistent Item because that's something that we need to explain what we actually do. It's not something where we have a list of different types of persistent item out there in the real world; we just know that Persistent Item is something that allows us to hang a whole load of stuff off in our functional requirement. So we’ve got these things because they're functionally required - we need the idea of Persistent Item to make things organised. We need an idea of a symbolic object and a propositional object in order to organise things; we don't have lists of symbolic objects and lists of types of propositional object out in our domain. But we need them in order to make types, appellations and conceptual objects hang together properly. <As for> some of the different ontologies that we've seen out there: there’s Dolce, which is from a lexical base, and uses intuition and processing of terminology. It’s a very good, theoretically motivated logical description, good foundational relationships, <but> has over-specified relationships in many very detailed modes of participation, which we find too heavy. It has a bad model of space-time, so it doesn't have the ability to talk about outer and inner bounds, it doesn’t have Allen operators and things like that. But apart from that it’s got a strong overlap with the CRM, so lots of stuff there. BFO - I can't remember what that stands for - is from a background of philosophy. It has a *very* poor model of relationships, it's about classifying things and it believes the world has a deterministic underlying reality, which we don't necessarily believe. What we're talking about is our observations of reality, not that there is some real truth. All we have is all the things that people have said about the truth, whatever that is, so we are observational-based and they believe there’s some reality out there that they're trying to produce an ontology for. It’s a nice idea, but it gets a bit scary after a while. It means that it’s very difficult to verify it, because there is never going to be any data about what they're thinking about, because it's “truth”. No databases are about truth, they’re about observations. Again, it has a strong overlap with the CRM. Then there is IndeCs and ABC Harmony, which are small ontologies: event-centric like we are, very strong overlap with the CRM and we’ve completely harmonised with them, so that the CRM has everything that they have embedded in it. We did a lot of work with the teams which were working on that, and they’ve agreed that the CRM now covers everything that they wanted to cover, they’ve stopped development and have just started using the CRM instead. Then there is SUMO, which is just lots and lots of concepts. It’s a very large, high-level set of concepts, and there’s no functional specification underneath it at all. It’s not very useful for doing stuff, but it’s a great place to get ideas – concepts that are out there.

## Slide 52

Another application is <that> we have developed a thinner version of the CRM and which gives us the ability to OII harvesting of CRM-capable datasets into a semantically rich subset of the CRM, a bit like Dublin Core is used to harvest data. But instead of getting the thinly specified Dublin Core data, we’re going to get a rich, event-centric set of data that we can harvest. So it becomes far *more* useful for doing semantic discovery, whereas <with> Dublin Core you end up just doing string matching and hope. Here we’re actually going to have semantically rich data that’s been harvested using this core technology. This is an example of things that we can do from record: <here> is some data about this bronze casting of Balzac which was created by Rodin, but he created the original figure and it wasn’t actually cast until after his death. So although it's Rodin’s statue, it’s Rodin’s statue which was created after he was dead. There’s that process, that complex process is in there, and this mapping allows us to see that that is available. So we can represent that complexity, but using a very thin core set of the CRM.

## Slide 53

And this is the core set of the CRM – it’s the concept of CRM Core. It has events, and all the information we need to drive events. It has the ability to drive events; it has the ability to categorise and classify things, identify things – give them names - carry free text descriptions; and it has the ability to relate one thing to another. So we have the ability to do “relationship”, “parts of”, “referred to by” – all of those things that we saw at those high-level categories right at the beginning of the lecture, when we looked at the different types of relationship we could have. So it provides us with all that functionality, and there are a couple of really good papers on that if you want to look at it in more detail.

## Slide 54

Then we can create minimal datasets using the CRM Core. So we can have an artist record in CRM Core talking about their events, and then we can relate that back to the work, the actual Rodin's original plan study and then the casting, through these work carriers – here are the Information Carriers. Here we say that here’s the identification of Rodin, <which> points back to the person, and then here's his death, and that points back to his death in the person records. We can harvest information very easily using an event-centric model, and link it together outside the original data repositories to provide a fast way of doing semantic discovery - a semantically rich discovery - rather than just the text matching that we’d get if we’d extracted Dublin Core records, where we wouldn’t really understand the relationships between events and the things that are embedded in them, in records.

## Slide 55

What are we trying to do with the CRM? Let’s have a couple of examples. Well, we want to get at semantic integration beyond context. What we mean by this is that I don't want to have to stay in the confines of the original data source. Within the context of the original data source, all the semantic stuff is there: I’ve got my desk instructions, I’ve got my thesaurus, I’ve got my process, I’ve got the software and the screen design that helps me put the data in. So within that, I’ve got a rich semantic interoperability within the data elements which are in my institution. But I want to be able to step outside that context and link those pieces of data to <other> pieces of data, to tell bigger stories from other institutions. The other institution’s data is in the context of that institution, I want to get at what's in those datasets without having to be an expert in that context, an expert in the way that organisation does stuff, and know all the desk instructions for how they filled in the database. I want to make explicit all the stuff that’s implicit in the way they work, so that I can join those datasets together. We’re going to try and pull together all relevant sources and data, so that we can do an evaluation of a question which wouldn’t be answered by just looking inside one database. I want to ask questions which are orthogonal to the way I record. Let's take an example: I’m going to record a whole set of paintings, so I will have a workflow which allows me to record paintings. I'll start off with one of set of paintings and I’ll go through and record all the information about the painting: the size of it, when it was painted, who painted it, what colours they’ve used and so on – what types of paint and so on. And then I’ll have another database in another institution, which is going a look at the biographical history of that individual. They’ll talk about when they were born, where they’ve travelled to, who they’ve met and so on. Within each of those, I've got lots and lots of information siloed in the context of how it was recorded. If I want to ask a question that’s orthogonal to that way of recording: I want to ask a question about what iconography on the paintings that this guy has painted is also present in works by other painters that he either met before he painted this picture, or before they painted the picture that has the iconography in. We can look at how particular images, particular ideas in the iconography have spread. That’s a really difficult question to ask within the silos that we have, because the information about where he travelled and who we met is in a completely different context to the information about the iconography of the paintings. So I need to merge those two different data sources in order to start asking questions which cross the boundaries between contexts. That's very much what we're trying to do: bring together all of that information from all of these different sources, and allow ourselves to ask these grand, super-arching questions that go beyond the boundaries of any particular data repository. We can do this using the CRM, and it gives us integration on a scale that’s usable. The Core allows us to do this in a harvesting way, so we can use the full CRM or extensions if we want to, but to maybe just get a handle on the problem and start the discovery, we can just use CRM Core and a harvesting mechanism to get us a long way down the line, and actually start to produce really interesting results without having to go to several institutions and completely blend all of their data. We can use the CRM as a blender, if you like, that sits on top of it. It allows us to do mediation across these different things. These different data sources, if we were trying to merge them, would all have to be very similar in the depth of detail that they went to, and the depth of specification that they had, whereas the CRM allows us to merge things which are of very different levels. I can work with a national level inventory, which has very little detail about things, with the detail of a very specialised research database which is going into incredible detail about one particular aspect of the thing that the general database is only skimming over. I can have – in an archaeological context - I could use a national inventory, and the research database of the palinologist who looked at the pollen from the ditches surrounding a particular burial site. And there, I’ve got incredible difference in detail. One’s just saying: “Hey, there’s a burial mound”, and the other’s going: “And in this little sample of soil, there was an oak pollen grain. Now that's a massive difference in scale and a massive difference in specification. But with the CRM, I can blend seamlessly between the tow. And that gives me enormous power - so I can deal with things like Dublin core and highly specialised databases at the same time. And by creating this common understanding, I don't have to go: “You will recall this way!” and “you will record this way!” so that I can make the match. I can just allow you to record and do the stuff that you want to do - follow your nose for new research questions, tackle new issues and still seamlessly blend your data to produce much richer stories across this massive range of things. So I don't tell you what to record, because you are the people that decide that. It's what we can then blend. As long as you're informed in your design decisions by the CRM, we can then blend your data with everybody else's and produce really rich stories.

## Slide 56

This is what we've just been talking about, in that when we do scientific and scholarly work we produce knowledge by adding to what's already there. We add new bits of research on, and it always comes in closed units, in documents - whether that's a database or an actual text doesn't matter, it’s a document - and they will have a history of evolution. The database will be enhanced, the document will be enhanced, new understandings, <there will be> new editions and so on. But the knowledge is very directed, it’s very in tune with what the original research question was. So if I want to spread out, it’s quite difficult from that. And it can only be evaluated in the context of how it was created and where it was created, as we have already discussed. What we want to do is take these document structures, open them up and make them explicit, so that they are not tied to their original context and we can get at the information in them and allow us to integrate them with other things. A good way of doing it is to have a creator, a creation place and a creation date. A bad way of doing it is to have a “provenance”. What does that mean? How does that work? We don't know. Or “place associations”. “This thing it is to do with Milan” – well, why? How is it to do with Milan? What we do about Milan? Is it that it was made there, is it that it was used there, is it that it was bought there? We don't know, so we want to understand how Milan relates to this object and life cycle date – that’s a *poorly* defined thing, whereas if we’ve got a good definition we can very easily map between the document and the CRM, and extract the information and make it explicit and usable. Of course, a particular data source might not be complete relative to all the things we’re interested in. And that's one of the joys of using the CRM: I can have a silo of data which doesn't have information about biographical details of the artist in at all, it just has very detailed information about a particular subset of his paintings. I don't have to have *every* painting in that detail; I don't need to know the biographical data in order to deal with that particular set of research questions. But I can hook them up with all of that information from other places and spread it out, so that I end up with a much richer story line. So any particular data stream doesn't have to be complete, it doesn't have to be rigorous, spreading out and covering the entire world. I can take these different silos and blend them together to produce much richer stories.

## Slide 57

The CRM really gives us three different attacks on managing the knowledge that we have in our environment. The first area is: how do we acquire the data, how do we grab stuff, how do we digitize things that we’re documenting? How do we get at that data? The acquisition of that data can be motivated by the CRM - we design our databases as schemas, so that we can gather data which is then exportable in the CRM. But the CRM is terrible at doing data input, because when we are doing data input we need ergonomic front ends to the databases. We need things which are going to help us go through the process. If I've got 50 vases to record, I want to be able to call up a screen for doing vases which has things about vases on, has specialised thesauri for vases and so on, so that I can make my descriptions of the vases very quickly; I don't have to think about all of the possibilities, I just need to concentrate on documenting that particular class of object. And it will have constraints, so that I can't say that this vase is 2000 metres tall – because we haven’t got any 2000 metre tall vases. So what we can say is: “What he actually means is 2 metres tall, he’s just put the decimal point in the wrong place” - those kinds of helpful things. It will use language specific to what we’re doing - it's pointless getting a pot historian to use art historian’s terminology to describe a pot, because that's not the way they think. It’s got to be optimised to the specialist needs of the particular user group that’s going to document things with it. They’ll often work on a set of items which are very similar, so we can do the “copy previous record into this record” and then modify it. If you’re doing hundreds of the same vase, you can document in that way and just change certain values in it. This will typically have very low interoperability needs for the actual acquisition; acquisition is usually inside a context, inside an organisation, inside a particular research project. But if we can design this so that we can map it, maybe even as part of the original design of the application, then the data becomes usable outside. And that's where our primary target is: the integration and comprehension of all these different data streams into [??]. Here we’re trying to break the document boundaries, get them out of context and merge them so that we can tell bigger stories. We want to match shared identifiers, aggregate alternative views of the same thing - <e.g.> the different fathers from different sources - and we don't care about which direction we’re going to ask the questions. So if I go to my vase database, I’m going to ask “How many red vases do I have?” *NOT* “How many vases made by so-and-so after this date include this piece of iconography and are red, blue or green?” So I have a much wider set of search choices in this richer environment, and this *must* be interoperable to be useful. That's what the CRM provides us with. And then at the end of it, the data that we've produced in this rich environment we can then use to tell our stories – to tell our history of objects, to look at the way people have used these objects over time, the evolution of these objects *and* we can look at the evolution of our understanding of these objects. So not only can we produce cultural history, we can produce cultural history of cultural historians. We can look at the archaeology of archaeologists, if you like, and understand how they’ve understood the past and how their understanding of the past has changed over time. So we can write histories of our profession as well as histories of the people in the past. And there we’re going to present stuff in order and give people the guide through our argument and our discussion: show them the evidence, and provides our audience with an understanding of how we reached our conclusions, as well as what the conclusions are. This is that of the end result, which is going to be based on the work we did in the rich CRM environment.

## Slide 58

One of the ways that we can use the CRM is an application is to do this kind of repository indexing, where we’re going to go into the individual documents - whether this is a database or a text document. We’re going to extract information out that refers to things in the CRM - actors, events and objects - and identify the fact that this is in this document, and this is in this document and it also appears in this document. So now I’ve got a link between this document and this document. And that process allows me to bring together these different silos of information and blend them into an overarching web of information, which I can run queries across.

## Slide 59

And this is the kind of thing that we can do: we can take digital library information from different institutions, and find out where this sample – who sponsored the expedition <during> which that sample was that gathered. So we’re going to leap across document boundaries, because the document boundary that talks about the setting up of the expedition doesn't mention the actual sample which was gathered in that expedition. In this case we’re talking about the discovery of Lucy, one of the very ancient type fossils. The records for Lucy don't include who set up the original expedition, but by jumping across documents we can say: “Ah! This institution helped in the discovery of Lucy, and that kind of thing can be very useful if you’re trying to look at the social history of an institution – what did they do? So that’s the kind of thing that we can do.

## Slide 60

Tony Gill was one of the development team on the CRM, <and> used the CRM to do this massive merge of different databases for the materials project at the research library group. From his work, he identified these five key points to using the CRM. First is that it’s elegant and simple compared with a comparable ER diagram, a comparable relational model, because the relational models tend to be very detailed and have to have different fields for every possible combination. It makes it much easier to analyze different… he had to analyze 1460 different data sources, in order to produce a single model which would enable all of these institutions to export their data and provide it as a single queriable platform. He found it elegant and simple; it allowed him to integrate information at varying degrees of detail. <For> some of the institutions this was just as a sideline database to their primary databases, so it wasn't detailed. At others, this was primarily what they were concerned with, so they had very detailed databases. He was able to use the CRM to bring them into line and allow them to integrate across all these different levels of granularity. It’s readily extensible by using the class typing and specialisations, so just by using good E55 Type thesauri, he was able to do a lot of the integration work without having to go into extensions, but the extensions were there for him to use if he needed to. In fact he didn't need to use them in that project, he was able to do it all by just managing it through types. The rich semantic content - actually picking out those embedded events and embedded identities, as opposed to names - he was able to draw inferences which he would never have been able to draw from the underspecified data elements in some of the databases. If he’d used Dublin Core, he would not have been able to do the things that he could do by using the CRM to merge these 1400 different data sources. It's been designed to do *exactly* what he wanted to do, so it was an absolutely perfect fit: it had all the functionality he wanted, but if he had used a different ontology, which didn't have that functionality in its design concept, he wouldn't have been able to achieve what he was able to achieve in a very short period of time. It was 1460 databases, and I think it took less than eight months from project start to completion, with the whole thing deployed. That’s an amazing achievement.

## Slide 61

What’s the state of development? Well, it’s ISO 21127, published in October 2006. The extension to cover the functional requirement for bibliographic records, and the functional requirement for something data… look-up data, master lists of people and titles, and the CRM has resulted in FRBR00, the functional requirement for bibliographic references object-oriented model, and that has now been approved by the International Federation of Library Associations, the owners of FRBR and FRAD, and CIDOC, so it’s now am out there as a public document available for use. What this deals with is thinking about works, about the intellectual conceptualisations. It takes a conceptual object and actually considers the entire history of it, from the moment you first thought of it through to its publication in a publication series, or as an individual book, and how to manage all of that, including handwritten copies and copies written… published by printing press or electronically published. So it covers the whole of the creation of abstract thinking, of immaterial ideas. We’re working with the TEI - the Text Encoding Initiative - in order to build up a model of how to look at the content of a text that’s been marked up with TEI, where the TEI shows us the structure of the document. So TEI gives the *structure* of documents, and CRM gives us content. We’re merging those two to produce a very rich markup environment for text, originally designed for mediaeval texts. We’ve got application models, like the CRM Core, and good and rich data formats, exchange formats and extensions, so there is ongoing work with different sub-communities who are looking at ways of pushing the CRM model out further, and making it more useful. For instance, there’s an OWL version, which is currently being finalised so that it will be available in the WCCC OWL language for use as an ontology out in that environment.

## Slide 62

So finally, what conclusions can we draw from this process? Strangely, everybody thinks that ontologies are very domain specific - that when you go for an ontology, what you're actually doing is really hemming yourself into working in a particular small environment. And what we’ve found with the CRM is almost the opposite - nearly all of it is *not* domain specific. I pointed out that very uniquely museum thing - the Current Permanent Location - but nearly all the rest of it is usable across a wide range of domains. And in fact we’ve been successfully deploying the CRM into medicine to look at the derivation and processing of cancer samples; <into> biodiversity, processing information about collecting samples and documenting biodiversity sampling. A whole range of different things use the same basic concepts that we use. So it’s a notion of scientific method - retrospective analysis, taxonomic discourse, which is common to archaeology and natural history - which show us that this model is useful *across* domains, it's not very specific to museums. We’ve been able to do it with a very small set of concepts. Everybody else has *huge* numbers of concepts, but we've got just over 200: 137 properties and 86 classes, and we been able to derive this massive range of things using this very small set of concepts. As we’ve added these dozens - indeed hundreds - of different formats to our understanding (we started from one standard, one CIDOC standard and we've grown out and we now have a couple of thousand standards that we’ve mapped), we’ve found that we add *very little* new stuff to the standard in order to explain these new richer formats for particular sub-domains. Some of them, getting very far out from the core of cultural heritage - the biodiversity stuff - we had to do virtually nothing to add all of the richness of biodiversity to it, just a very small extension. We’ve made the investment, it pays off and we think the CRM should be our language for semantic interoperability, certainly within the cultural heritage domain, but we’re finding that we can spread it out into natural history and indeed into some of medicine as well. So we’re really finding that the CRM gives us a *huge* capability to move data around between different groups, and that it gives us semantic richness to that. It's not just doing pattern matching in the text that we produce, we actually understand what that text means, and we can get at that meaning and relate it to other things. Thank you very much for your attention. I hope you've enjoyed this presentation, and we look forward to you using the CRM. Please tell us when you’ve used the CRM for your project, because we would like to keep not a handle or a control on it, we’d like to see all the different ways it’s being used, so that we can explore how to use it in further circumstances. Thank you very much.

Also available: <http://www.cidoc-crm.org/cidoc-crm-tutorial>

**Application Development:**  
Centre for Cultural Informatics  
Information Systems Laboratory  
Foundation for Research and Technology  
  
**Project Team**  
*Scientific Coordinator* : Martin Doerr  
*Project Coordinator* : Chryssoula Bekiari  
*System Design*: Martin Doerr, Christos Georgis, Chryssoula Bekiari, Manolis Daskalakis   
*System Implementation*: Manolis Daskalakis  
  
**Authoring and Presentation**  
Stephen Stead, Archaeologist, founder member of CIDOC CRM  
  
**Graphic Design**  
Maria Christaki, graphic artist  
  
**Text editing**  
Ben Petre, Linguist  
  
**Director of Video**  
Michalis Serpetsidakis -  
MIND Film & TV Productions  
  
**Editor of Video**  
Kostas Dimakis  
  
**Copyright***CIDOC CRM, 2008*