

4

Developing the Common Metadata Model (CM²)

4.1 Chapter Overview

This chapter begins by describing the role of the common metadata model (or CM²) as providing the most stable foundation for the development of future service-based architecture (SBA) components. SBA components may be defined as the intersection of a specific business functionality and the subsets of the organizational technology and data architectures providing support for that function. A concrete example might result in the scope of an architectural component being defined by the business as the business processes, the technology and the data supporting organizational benefits operations. This architecture could be described as simply as the "PeopleSoft™ version 7.5 benefits module implemented on Window 95."

The CM² architecture provides the guiding framework for development of SBA components. The quality of the SBA components developed cannot be greater than the quality of the CM². Following this description, we present a sample scenario that is used to illustrate the SBA development concepts presented in this and the next three chapters. The sample scenario – based on real life examples – illustrates typical aspects of a CM² project scenario. This chapter closes with a brief description of CM² development critical success factors.

(As a quick read ahead, Chapter 5 next describes data quality engineering aspects specific to the development of the CM²; Chapter 6 describes the analyses required by the archeological transformations to produce the facts required to develop future architectural components and the analyses required by the archeological transformations that use the archeological facts to produce new architectural components; and Chapter 7 describes coordination required to develop the component integration framework – that architecture development activity that creates tangible business value.)

4.2 Role of the CM² within Architecture Development

The CM² has been something of an architectural grail. At the time of this chapter (February 2001), most organizations do not have a CM². Instead, they have one or more existing systems that are not well understood in terms of data, process, and linkage to organizational strategy. More sought-after than achieved and most writing about – it, to paraphrase popular management quote:

[m]ost data architecture planning is like a rain dance. Most discussion focuses on the style of architecting (dancing) and

**not on the whether it produces better business results
(whether the dancing makes it rain or not)!¹**

Unfortunately our research has detected a similar pattern: more CM²-type projects have failed than have succeeded. We have each had CIOs look us in the eye and remark "Yes, I know that data-centric is the way to develop applications, but have you seen one that worked?" We have seen many examples; however those organizations that do a good job at data-centric development generally consider its use a strategic advantage and closely guard their methods and successes. Unlike Polo Ralph Loren, other organizations don't always like to describe the means by which they stay ahead of the competition.

We have assumed that after reading the first three chapters, the need to evolve toward SBA would be evident. The question most frequently occurring at this point is something like:

"Sounds terrific and practical but I'm forced to migrate from a complex environment. Where should I begin and what blueprint can I follow as I perform the detailed engineering that is required?"

This reasonable question hides an underlying layer of complexity that has caused much confusion about initiatives of these types. This complexity is illustrated in Figure 1 (below) that shows the role of the common metadata model (CM²) in the development of an organizational SBA.

¹ The original quote: "Most discussion focuses on the style of dancing ... not on the weather!" was written by Russ Ackoff in the Strategic Management Journal article titled "On use of models in Corporate Planning" 2:353-359 (1981).

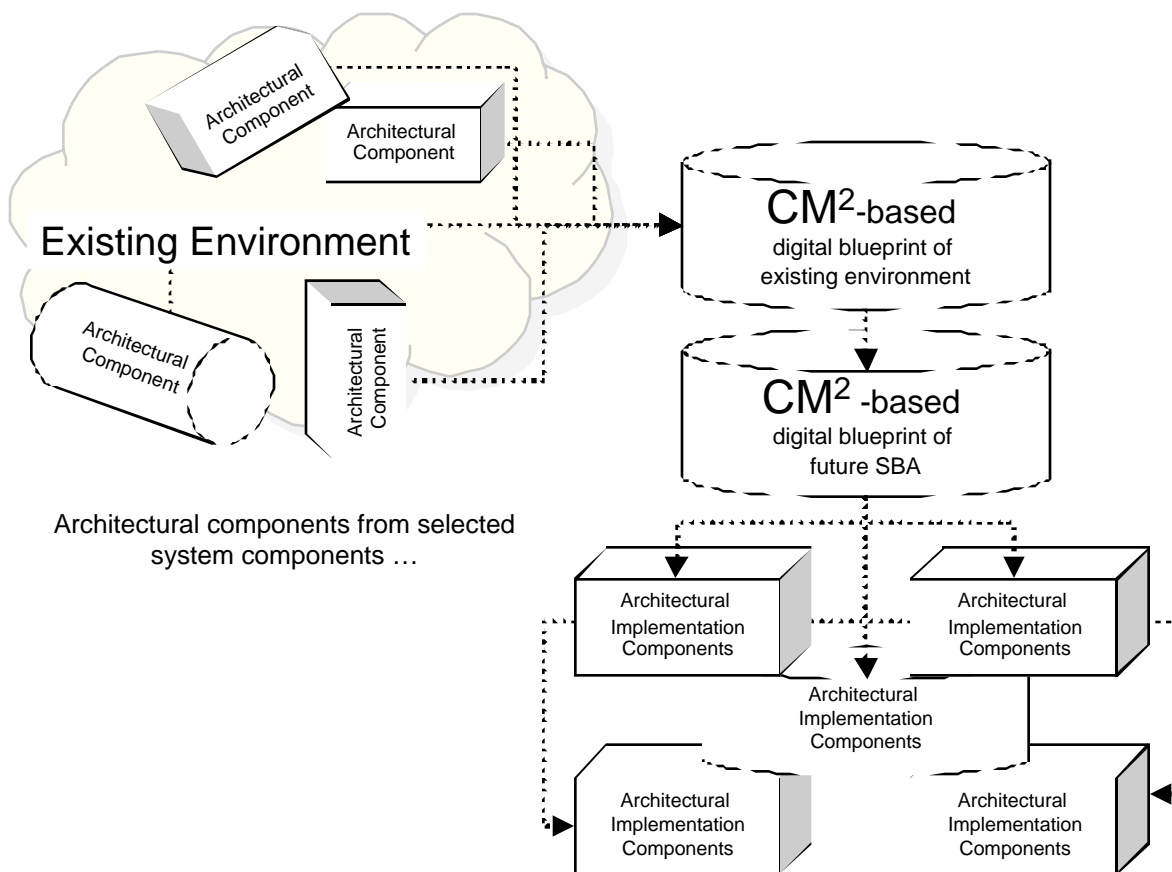


Figure 1 The complexities of "just getting started."

Because organizations adhere to different organizational strategies, they often require specific SBA components developed specifically for the organization. Before an organization can develop an SBA capable of supporting organizational strategy, it must first understand its existing system architectural assets.

4.2.1 "Understanding" in the Context of SBA Development

We define the term "understanding" very specifically in the content of SBA development.

- Understanding is a shorthand reference for use of a model-based technique to represent, manage, and develop manipulable system model components within the context of a formally defined SBA.
- These model components use standardized notation and are sufficiently detailed to permit both business analysts and technical personnel to separately read the same model, and come away with a common understanding.
- Crucial to achieving this level of understanding is that both share a fundamental understanding of the business environment in which they work.

- Development of system components using model-based technologies permits development of objective measures of progress and quality.

In this way the model becomes the basis, the language, the currency for achieving the understanding not only among project team members, or between organizational technologists and business knowledge workers but also among all organizational personnel. Some examples of commonly understood technical and business information that is used are included in the following example. Consider the following definitions that are also illustrated in Figure 2 (below).

- 1) Each FACT combines with one or more MEANINGS.
- 2) Each specific FACT and MEANING combination is referred to as a DATUM.
- 3) An INFORMATION is comprised of one or more DATA that are returned in response to a specific REQUEST
- 4) INFORMATION REUSE is enabled when one FACT is combined with more than one MEANING.

Figure 2 is a model which when combined with the definitions in the paragraph above, permit different individuals to share semantic understanding of some types of detailed technical and business information. Both parts of the model—definitions and graphical relationships—are required for complete understanding. Now that both readers and authors have a shared understanding of "understanding," references to the term will be understood.

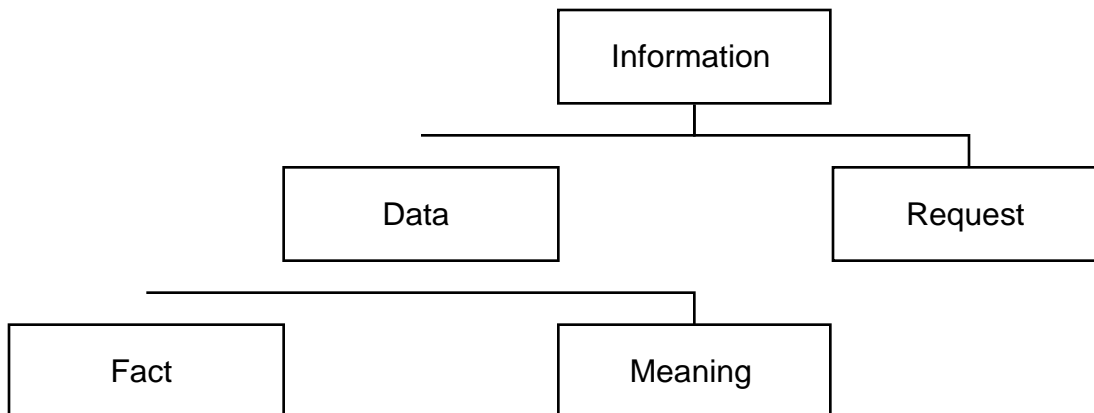


Figure 2 A model precisely defining the relationship between the terms INFORMATION, FACT, MEANING, and DATA.

4.2.2 Architectural Discussion

As shown in Figure 3, the CM² is an integral requirement of optimal architecture development.

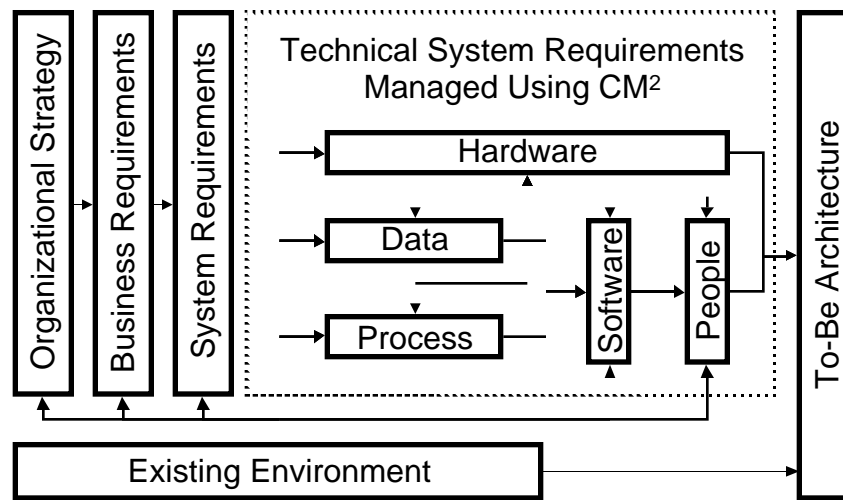


Figure 3 Optimal Requirements Flow for Architecture Development

When considering architectures as guiding the transformation of strategic organizational resources needs into specific information systems development projects, it can be said that all organizations have architectures. An architecture that is not articulated and/or understood cannot contribute to organizational success. All systems have architectures. The information represented by the architectures is more valuable when maintained in a format useful to other organizational activities. Again the CM² provides a means of developing and evolving the SBA components. Prior to an organization formally expressing its architecture as a CM², the architecture does not exist in a tangible format, making it more difficult for others to assess and access its contents. One of the chief architectural functions is to facilitate the interaction among organizations, individuals, and systems.

Architectural development is challenging work that involves art, skill, and intuition as well as detailed technical component engineering. To optimize development processes, the information contained in the integrated metadata collection, that is the CM², evolves into the information that comprises, articulates, and denotes your architecture. Successful architecture development may be said to require the following steps, which must be applied-in the correct sequence-with the proper inputs-from the existing environment.

- What is the organizational strategy that the architecture is being developed to support?
- What are the business requirements that this architecture must satisfy?
- What are the specific system requirements that the architecture must support?

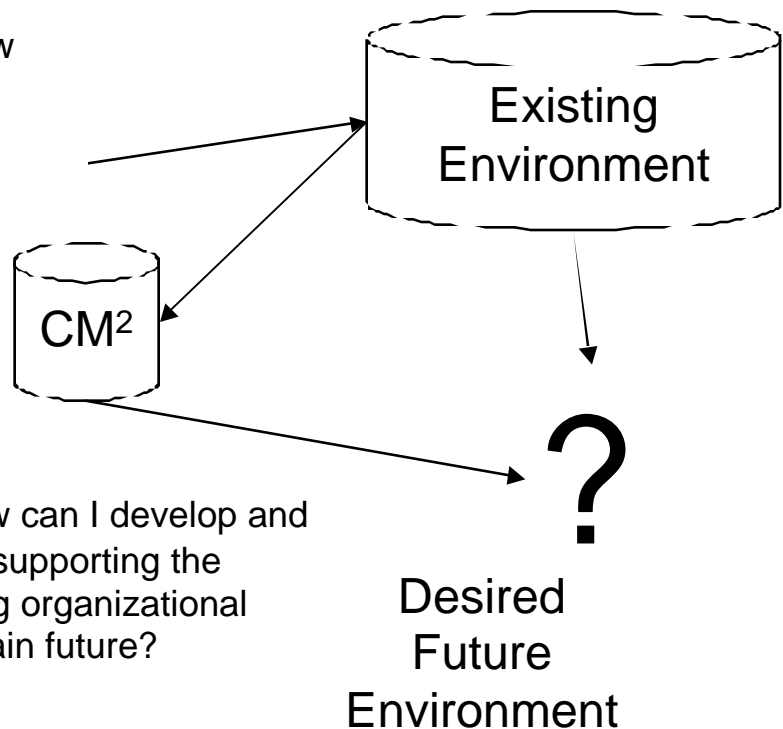
Once the system requirements are understood, development of the technical system components can begin.

- Development of hardware requirements may be driven by existing organization investments or may be postponed until after other system components are specified.
- Development of organizational data and process requirements using the same advanced information management technologies that we use to manage organizational data.
- Once the correct data and the process information is programmatically manageable, software and people component requirements can be specified and integrated with the data and process requirements.
- What is the timeline for implementing the architecture, and does obsolescence become a factor for either legacy or proposed environments?

When components have been specified, the desired architecture can be developed. Only after all of the above steps have occurred can the organization begin to realize benefits from the newly developed architectural components. All these components must be specified in a form useful to multiple levels of organizational users. Optimal organizational architectural support can be developed when this information is commonly understood by both business analysts and IT staff. While it sounds simple enough, maintaining detailed information such as what business information is required by what business processes, becomes a key organizational asset.

The CM² development process ensures that critical information is available when needed so that development of the new system components is as effective and efficient as possible. A two-phase development process is required unless you are developing new systems from scratch, in which case the first phase can be skipped. Each phase is designed to provide key information essential to making subsequent decisions regarding the development of new system components. In macro terms, there are two primary questions that must be addressed to develop the basis for the architecture as illustrated in Figure 4.

Q1) Archeologically speaking, how should my understanding of my existing environment best inform/influence the selection/development of new components?



Q2) Architecturally speaking, how can I develop and arrange components capable of supporting the business needs while maintaining organizational dexterity in the face of an uncertain future?

Figure 4 CM² Development Process Reinforces Data Engineering Principles

Savings accrue: from CM²'s common metadata format because component metadata does not have to be recreated each time it is required; from improved organizational communication and activities resulting from model-based dialogues that incorporate component metadata; from better business engineering practices resulting from rapid access to integrated process and data metadata.

The CM² development process takes one through the steps required to develop a logical, normalized, integrated (between process and data) model of components of the existing environment. This logical, existing model will provide the basis for answering question two (above). Typically question two cannot be correctly articulated unless the answer to question one is known.

4.2.3 CM² - A Digital Blueprint

The CM² functions as a digital blueprint – providing a structure and format for storing metadata that is required to develop and implement future SBA components. The CM² provides the digitally based common metadata required to transform existing information infrastructure into new information architecture. ("Infratecture" likely describes the existing system environment as a brittle unarchitected collection of systems.) CM² stores metadata in a format where it can be used to develop SBA components by organizational activities (in particular data architecture, business engineering and strategy development).

Perhaps as difficult as managing the existing technical environment has been the task of explaining the required approach to CM² development to management.

The process of developing a CM² is accomplished through a series of iterative builds. Each build results in a refined component replacing an existing component or a new component being added to the existing CM².

As shown in Figure 5, over time, organizations develop a CM² by reverse engineering selected components of the existing environment in order to recover desired architectural components and other valuable information from the existing system(s). In each instance, a specific existing component is defined, identified, located, reverse engineered, and transformed into an existing architectural component. The components are added to the CM² and integrated with organizational strategy.

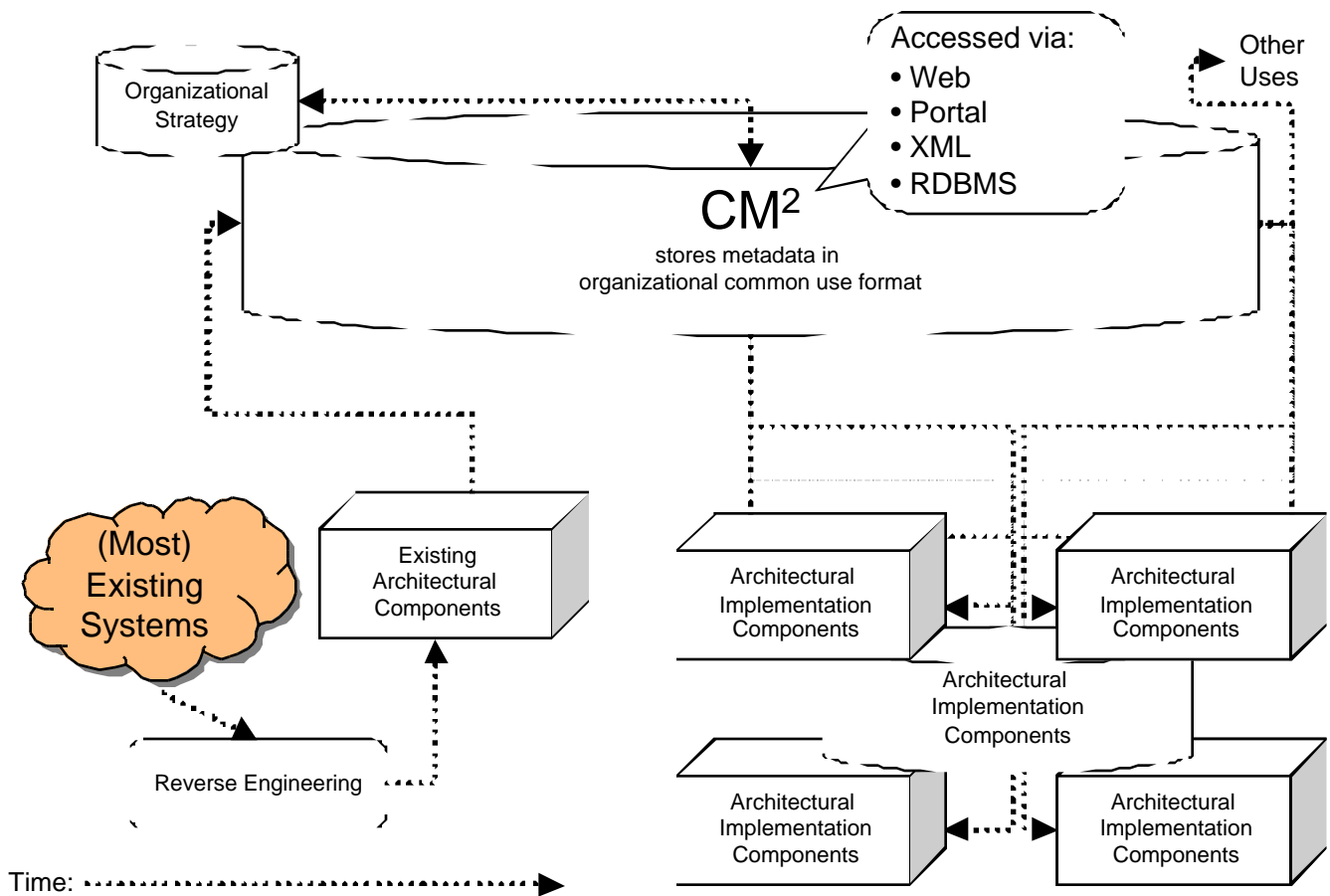


Figure 5 Role of the CM² within Architecture Development

While not rocket science, CM² development does present a specific series of technical challenges that require a combination of art and science in order to properly address _____. For example, it is often very difficult to determine in advance what will make a “good-sized” system component. In advance these system components are referred with the descriptive system term, “chunks.” The combination of architectural skills required to develop a comprehensive, integrative approach while utilizing well-engineered components has not been easy to comprehend. Engineers typically do not create architectures and architects typically do not engineer components for their **creations**. Both types of inputs are required for

development of the new architectural components. It has been challenging to assemble CM² development teams with the correct blend of top-down and bottom-up perspectives.

Perhaps a useful parallel can be drawn by stating that metadata architecture is driven by, and serves much the same purpose, as civil architecture – to create and develop technical systems too complex to be treated by engineering analysis alone. Like other semi-soft topics, business value of data architecture engineering has too often been not well perceived by those who are requested to fund such investments.

CM² architectural components are delivered with structures and access means (Web, Portal, XML, RDBMS, etc.) that facilitate and encourage its use throughout the organization. CM² is used in organizational data architecture activities to development data architecture components that support implementation of organizational strategy. Business engineering teams use CM² to define and align business practices with organizational strategy. Strategists use CM² to instantiate specific strategies with organizational data and practices. All access integrated process, data, and strategy metadata using the means that best supports its use.

As an integrated collection of metadata, an organization's CM² should mirror the current implemented environment. Different versions of CM² correspond to and eventually become synchronized with organizational system component implementations. CM² components are shared among organizational functions. CM² components function as both architectural specifications and management technologies. The architecture is expressed as the structure of the CM², and the CM² is used to implement systems components within the architectural context. CM² also functions as the source of the data used to subsequently implement the integrated organizational environment. As the components are implemented, data stored as part of the CM² is used to develop architectural and physical component structures.

4.2.4 CM² Metadata Commonality

Development of a CM² is common to many IT investments, hence the term "common metadata model," whether the task is to:

- begin the process of embracing an enterprise resource application package (ERP I or II²);
- reengineer one or more existing systems in anticipation of e-business demands;

² Gartner now describes ERP II as "thereby rendering the vast majority of current ERP systems obsolete from both architectural and business relevance perspectives," and looks forward to ERP II as "the second vision for ERP I." – <http://www.gartnergroup.com/> accessed on 12/4/2000.

- attempt to achieve IT cost saving by modernizing and harmonizing the organizational computing capabilities;
- invest heavily in infrastructure as a defensive measure against a forecast economic downturn; or
- (having been blessed with enlightened management) simply attempt to better position the organization for the future by implementing component capabilities ranging from gaining operational efficiencies to new strategic sources and uses of information.

CM² development is common to all of the tasks listed above (and many more). Collective questions have been similarly worded.

- How do we get started?
- Must I examine every system in detail?
- What method should I follow?
- How can technology help me?
- How long will it take?
- How much will it cost?
- Where and how do I apply data quality in all of the above in order to achieve quality enterprise data?

The last question will be addressed in the next chapter. It describes data cleansing/data quality issues that must be addressed before full use of the new CM² can be achieved.

Contextually, the challenge can be described as follows. (Take a deep breath!)

Understand the existing environment well enough to:

- describe your existing environment as a series of discrete, identifiable components;
- correctly identify the Pareto subset – that is, understanding the entire environment well enough to identify those components that can provide the highest payoff, those components which can tell us the most about the environment;
- know when components are modeled completely;
- know which components to keep;
- know which components to understand well enough to re-implement internally using modern technology;
- know which components to outsource;

- know which components to stop supporting;
- know when you have modeled enough components to achieve critical mass – satisfying the Pareto hypotheses;
- model the components of the existing business processes and technology supporting them; and
- effectively discuss the model correctness with management and improve its correctness based on feedback.

Understand systems and business engineering tools and techniques well enough to know what technologies to invest in for your environment when:

- selecting, investing in, and applying technologies required to identify and extract relevant information from the existing environment;
- developing and utilizing a new form of metadata repository capable of supporting your organizational needs for CM²-based information; and
- extracting and formalizing organizational knowledge so the right CM²-based information is available to the new system developers it is required.

Achieving understanding of the business' technology requirements:

- The strengths and weaknesses of the business capabilities supported by the current environment;
- Where organizational strategy is anticipating the organization will evolve in the next phase of its existence;
- By both organizational management and IT management of what IT-based support is required and what resources will be required to provide that support;
- **Develop and populate the new architecture.**
 - Design and implement the CM² in a (usually phased) manner capable of supporting the identified and unanticipated business requirements articulated in the organizational strategy.
 - Migrate suitable components into the new environment.
 - Engineer the transformations required to modernize existing components.
 - Implement application engineering so that unanticipated organizational information needs can be implemented using a common delivery technology.

- Ensure that the new environment has data quality controls engineered into it so that existing data quality problems do not infect the new environment.
- **Implement the new CM²-based components.**

The list represents the core technological basis of many IT investments. Failure to address these requirements has been at the root of many project failures.

4.3 ALC Sample Scenario

As mentioned above, organizational CM² tends to provide organization-specific support. Understanding an organization's CM² is understanding both that organization's strategy and its technology plan to carry out the strategy.

Consequently, being constrained against revealing technical details of the Polo SBA to the competition, this section instead introduces a sample CM² development scenario for a large company (we'll call them ALC) that illustrates the concepts covered in the next three chapters. The example begins with an e-mail message from a colleague we refer to as A. Data Architect. Figure 6 (below) shows an e-mail received by one author during the time these chapters were being written.

From: adataarchitect@alc.com
Date: Tue, 12 Dec 2000 15:40:53 -0600
Subject: Details for Statement of Directions
To: paiken@acm.org

Peter,

Attached is the document that describes the projects that we would like statements of directions proposals for. I encourage the statements to be direct and to the point and answer the WHAT and HOW questions. That is important to ... (our management). Don't want to beat a dead horse, but the concepts have been sold to anyone that matters here, we need to deliver now.

Talk to you later.

A. Data Architect

Figure 6 E-mail containing typical CM² development questions

A few interesting points are made by A. Data Architect in the message. First of all, it clearly points out the type of dialog that was occurring between A. and Peter Aiken. Peter, thinking he was being helpful, had been sending materials to A. that he thought would help A. justify the development of ALC's CM². A's response indicated the belief that management had a good understanding of the positive results that would accrue to ALC as it evolved its existing unarchitected environment into something both more supportive of ALC's strategy while being simultaneously less expensive to support. A's e-mail message also illustrates how management wants to understand a specific procedure for recovering the infratecture-base components

from the existing systems and evolving them into a more useful architecture-based form of support.

4.3.1 Facts About the Existing Environment Known at Project Inception

Preliminary work had been done by ALC to characterize its environment and attempt to determine the size and specifics of the challenge faced in developing a new architecture. The existing environment consisted of three types of data management technology. Figure 7 describes the measured characteristics of the existing environment at the time of the e-mail message – an environment that should seem familiar to many readers.

Technology	Logical/ Virtual Databases	Subject Areas	Tables	Attributes	Unique Attributes	Records	Programs/ Copybooks	Lines of Code
Data Management Technology Type 1	44	20		13,067	6,600			
Global schema						1,049		
T01							5	
C01							4	
Data Management Technology Type 2								
R2			613					
R3			108					
R5			127					
R7			447					
R72			5,996					
R73			11,224					
Data Management Technology Type 3			1,227	9,000	2,514			
Application							10,970	15,700,000
Copybooks							5,518	498,966
Totals:	44	20	19,742	22,067	9,114	1,058	16,488	16,198,966

Figure 7 Measured Characteristics Of The Existing Environment

The first data management technology was a rather old (by February 2001 standards) network style DBMS. The systems managed by the first type of data management technology were estimated to contain 44 logical databases supporting 20 subject areas (such as order management, warehouse management, product design, etc.). Other known characteristics of the environment included 13,067 data elements with just over one half (6,600) being unique. The number of programs, copybooks and lines of code was unavailable.

No one had developed a reader or port technology permitting automated or semi-automated extraction of the metadata for this technical environment. A technology vendor had expressed interest in collaboratively supporting this project because it was interested in co-developing a new port technology for them. If it came to fruition, this could have a major cost impact.

The second data management technology was implemented using a popular RDMBS largely in support of an enterprise resource package (ERP) that covered two more functional areas – HR and Financials. The seventeen thousand tables supporting this implementation accounted for 93% of all data tables in the existing environment. Also significant, the organization had developed some other

applications, the precise function of which still had to be articulated. These applications were supported by almost 1,300 tables—the data structures of which were largely unknown. The precise number of attributes and unique attributes was unknown for any of the tables supported by the second data management technology.

The third data management technology was one that closely competed with the second data management technology. More than 1,200 of these data table types had been developed to support a client-server environment. The tables contained approximately 9,000 attributes, and more than 2,500 had been identified as unique. More than 16,000 COBOL and Assembler programs and copybooks organizing more than 16 millions lines of code had been implemented in support of the third data management technology.

The metadata of all three data management technologies must be understood because A. Data Architect had identified all three types as playing a significant role in future SBA components. Collectively, the challenge for analyzing the data-based components of the existing environment amounted to at least examining almost twenty thousand tables. Happily, much of this metadata already existed in the mainframe-based COTS repository. This daunting challenge was perhaps made more palpable with the understanding that the ERP metadata could be recovered by programmatically reverse engineering the ERP system's data [Aiken 1999]. Still this left more than twenty-five hundred tables and nine thousand attributes to be examined by subsequent analyses.

Notice it was impossible to determine which of the three data management environments presented the greatest understanding challenge. At this point in time it was impossible to objectively compare the various data management technology environments. No standard measure could be applied to them.

4.3.2 *Project Definition*

Four specific tasks identified with deadlines were:

- network database reverse engineering (within the next 90 days)
- data classification (within the next 90 days)
- client server software reverse engineering (within the next 90 days)
- preliminary metadata architecture development (within the next 5 months)

ALC management indicated short-term deadlines for the first three tasks and that "big bang" deliverables toward the end of the deadline were unacceptable. Accordingly, project planning was geared toward regular deliverables—about 1/3 every thirty days with interim deliverables as appropriate. Each task would produce deliverables that would represent one or more identifiable, integrated CM²

components. Each CM² component represented aspects and attributes of the existing environment that were required to develop the desired future SBA components.

The first task was to refine the numbers that were catalogued in Figure 7 and to fill in the blanks for missing cells on that worksheet. This was accomplished during the preliminary system survey and, while it didn't reveal any particularly astonishing numbers, the fact that the metrics would be filled in and verified helped to reduce the project risks associated with unknown system characteristics. Next, candidate components were identified in each data management technology. Metadata describing selected candidate components were extracted from the existing systems. This information was integrated into CM² prior to most of the metadata architecture development work taking place.

Delivery of major system components (defined as an "initial architectural structure") would be delivered in 90 days. The project team would deliver major project deliverables according to the notional schedule of work plan (Figure 8) below.

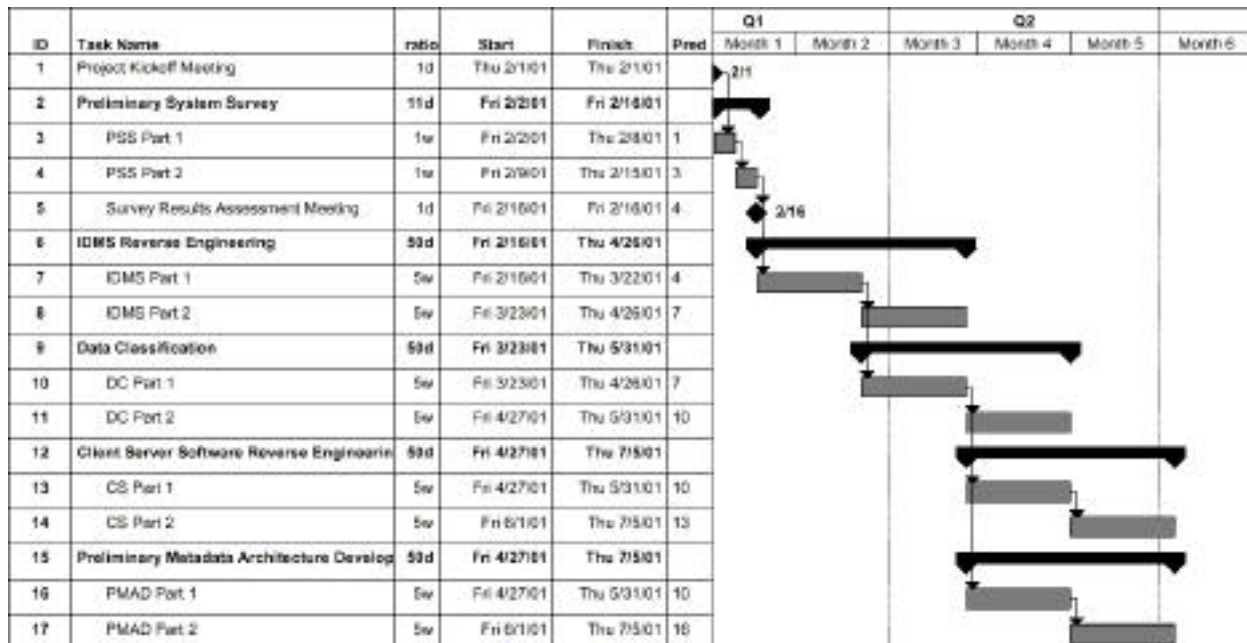


Figure 8 Sample Scenario Project Plan

Management desires a 90-day deadline for tasks 1-3 and that "big bang" deliverables toward the end of the deadline are unacceptable. Accordingly planning is geared toward regular deliverables – about 1/3 every thirty days with interim deliverables as appropriate. The project team will deliver major project deliverables according to the notional schedule of work plan below. At this level of project understanding it isn't useful to distinguish beyond the start and completion of the first half and second half of individual tasks. Each of the four tasks has been assigned ten weeks of resources. Subsequent tasks may often begin when a preceding task has reached critical mass – usually about halfway through the project. In the example shown the first two weeks of the project are spent refining this original

estimate. The IDMS task begins next and when it is approximately one half complete – about five weeks later – the data classification task may be started. Once the data classification task has reached the fine-tuning stage, the client-server reverse engineering and metadata architecture development tasks can begin.

It is important to note that, while the phases are shown sequentially, each phase is tied to the size and complexity of the component that is the focus of the archeology-based transformations. (This last sentence is just another way of saying that, right now, we don't know enough about the project so we have just provided a standard template for the project.) Development of the actual CM² and SBA component occur in a manner that has been hard to describe. The CM² was developed over a series of iterative developments, each iteration refining an enlarging, comprehensive set of metadata components. Analysis and transformations were limited to component size—which becomes a key variable. Components are also the units of parallelism with respect to throughput. This is illustrated in Figure 9.

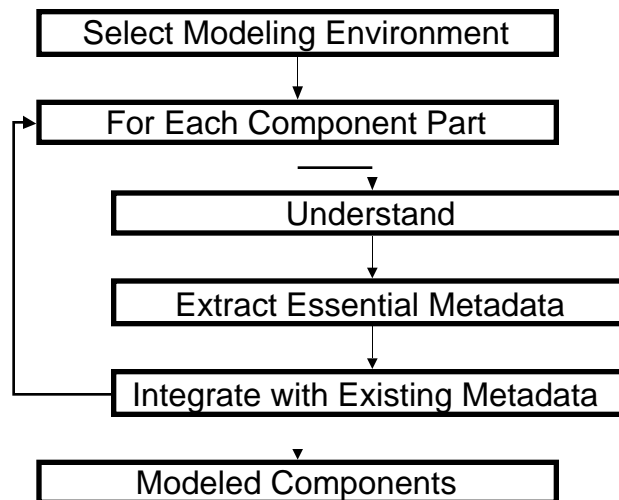


Figure 9 Implementation -> Model

Key also to this approach is selection of appropriate components. Each component progresses through a two-phased evolution consisting of eight required transformations. They are illustrated in Figure 10 (below).

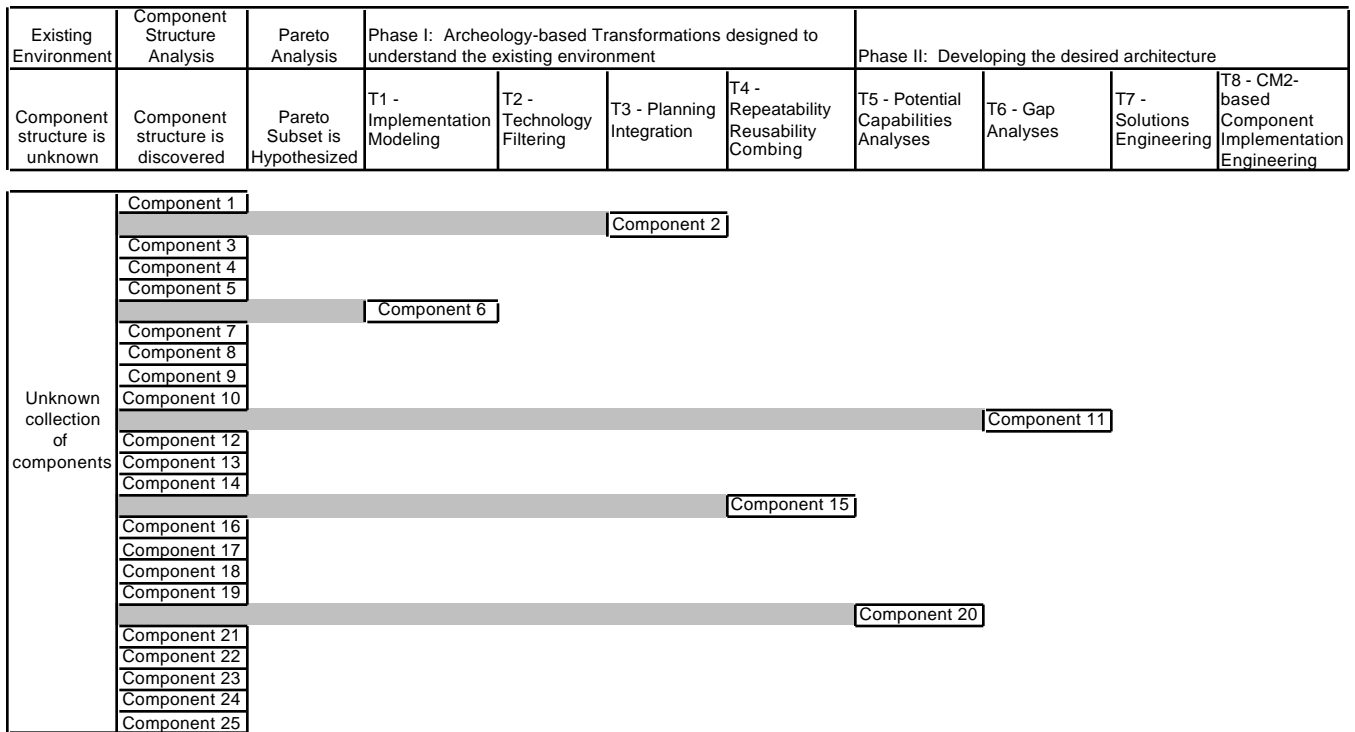


Figure 10 Sample components progressing through the transformations

Please note that the figure makes no attempt to characterize the relative size of each component. A variety of scientific visualization technologies can assist CM² developers with this aspect of understanding.

In the sample scenario, an existing technology environment (such as #1 [?]) might be initially largely unknown. A first step would be to conduct a component structure analysis that reveals actual components or “componentizable” structures. In the example, this analysis yielded a total of 25 components. The next analysis attempts to determine likely candidates for the Pareto subset – those 20% of the components which, if fully understood, will provide 80% of the critical mass of understanding the component collection. The five components (20%) identified by this analysis are component numbers 2, 6, 11, 15, and 20.

The figure also illustrates that each component can experience the transformations with different rates of progress. The first component (component 6) might be modeled first because of its central importance. The information from the T1-based model yields information that facilitates analysis of the other Pareto subset components (2, 11, 15 and 20). The figure shows that component 6 has completed T1, while component 11 has completed T6, etc. The Pareto analysis leads to the hypothesis that these putting these five components through the two transformation phases will give a solid basis to the CM². Once the five components have progressed through the entire set of Phase I transformations, it can be determined whether sufficient mass exists to begin architecture definition.

This approach must be implemented for the other two data management technologies.

4.3.3 Estimated Resource & Project Team Requirements

Internally, the project can be accomplished on a time-and-materials basis according to the following table but subject to the following disclaimer: *the results of the preliminary survey might redefine the project costs by a factor of as much as 50%*. The reason for the disclaimer is that until the preliminary survey is complete no one has enough information to produce a truly useful resource estimate.

Please note: The fifty percent variance is based on more than a decade of CM² development experience across more than twenty-five sizable projects.

The project team will develop refined and detailed task/cost breakdowns once the preliminary survey is complete. For the sample scenario the estimates produced from the preliminary survey are illustrated below as Figure 11.

	Category	Number	Percentage of Total Project Cost
1.0	IDR Staff		
a	Project Direction	2	15%
b	Task Management	2	10%
c	Senior Technical Specialist	1	2%
d	Senior Research Associate	2	20%
e	Research Associate	4	16%
f	Research Assistant	2	6%
2.0	Software		15%
3.0	Travel		15%
		Total:	100%

Figure 11 Initial Planning Numbers

One means of addressing project risk is the use of the preliminary survey. The survey identifies the areas of highest uncertainty and project risk. These are quickly subject to rigorous analyses aimed at reducing the risk. The answer to the question:

"How much will this cost and how long will it take?"

is

"It will cost at least as much as the preliminary survey."

The goal of preliminary survey is to provide a low risk quote for the entire project. The preliminary survey will identify the total work required, determine what tools are available, and assess whether this situation will permit desired results by the completion date. The scope of the preliminary survey will vary according to

the specific situation. In some instances expanding to include business rules and/or reports and process reengineering and in others focusing on just a few key architectural components.

Based on many years of performing data engineering analyses for dozens of organizations, the size of the project team was determined primarily by the project time frame (3-5 months), the number of parallel teams proposed (2), and the availability of data model apprentices from a university spin-off organization specializing in data engineering. It was determined that the use of parallel development teams would facilitate throughput. Figure 12 describes a project team consisting of 13 team members according to job categories, estimated total project hours, and total project days.

Category	#	Semester Weeks	Semester Hours/ Week	Summer Weeks	Summer Hours/ Week	Project Weeks	Total Hours
Project Direction	2	8	15	3	15	11	330
Task Management	2	8	20	3	20	11	440
Senior Technical Specialist	1	4	20	0	20	4	80
Senior Research Associate	2	18	40	3	35	21	1650
Research Associate	4	18	20	3	20	21	1680
Research Assistant	2	18	20	3	20	21	840
Totals:	13	18		3	35	21	5083

Figure 12 Project Team Commitments

Please note: Use of apprentice-based modeling team skewed the project hours in a manner that is different than if a full time team had been dedicated to the project. A dedicated CM² team may be expected to produce at a higher rate.

Key to the throughput on this project was rental of a key data analysis technology from Evoke Software. The Axio technology is based on a powerful inference engine that allows data engineers to quickly form candidate hypotheses with respect to the existing data structures. These hypotheses are then presented to the SMEs (both business and technical) who confirm, refine, or reject them. In this manner the technology allows existing data structures to be inferred at a rate an order of magnitude more effective than previous manual approaches. Usage costs for the 21-week project period were projected to be approximately 15% of the project budget.

4.4 CM² Development Critical Success Factors (CSFs)

Our discussion of CM² development closes with a brief description of CM² development critical success factors. Much of the success is attributable to the CM² **developer**, the IT professional who is responsible for ensuring that the steps in the

development process are followed. Several success factors that have been critical to the success of CM² development are explained below. They include:

- use of structured techniques
- understanding of the interdependent transformation phases
- correct implementation of cm² components versioning
- securing quality subject matter expert participation
- promoting an awareness of cm² development timelines

Data quality as a SBA component is addressed in the next chapter.

4.4.1 Use of Structured Techniques

When we use the term *structured techniques*, six specific characteristics³ permit CM² developers to understand the problem, be guided to a solution, and enhance assessment of the results and payoffs in terms of organizational knowledge and investment. Benefits of using structured techniques follow.

1. *Allow the form of the problem to guide the form of the solution.* The individual characteristics of existing environments tend to make each CM² development challenge rather organization-specific. Characteristics of the sample scenario led to the development of the new architecture. (Recall Maslow's observation: "If the only tool you know is a hammer, you tend to see every problem as a nail.") Structured techniques are useful because they are adaptable to meet both changing understanding of requirements and increased problem understanding.
2. *Provide a means of decomposing the problem.* Problem decomposition has long been a part of structured techniques. To *decompose* something is to separate it into its components or basic elements. *Decomposition* is the process of dividing up large problems into smaller, more manageable problems. In this fashion, a solution to the large problem can be developed by solving the smaller problems. For CM² development, our vague cognizance of each system must be transformed into understanding of each system component. (For example, each component is rated according to whether it contains much detailed information (plus), average information (), or no information (-), etc.)
3. *Feature a variety of tools simplifying system understanding.* Often these are graphics-based, permitting those who are not familiar with modeling

³ The definition of structured techniques is derived from one presented by Alan L. Eliason, *Systems Development Analysis, Design, and Implementation*, Scott, Foresman and Company, Glenview, Illinois, 1990.

tools and terminology to quickly become engaged in the process. The various Office-based technologies required to implement the CM² are considered standard parts of the knowledge worker tool set.

4. *Offer a set of strategies for evolving a design solution.* Like management, architecture development is an art as well as a science. This means to achieve the desired results it is necessary to understand the situation in the context of long-range goals and draw on appropriate tools in response to evolving situations. The key to effective implementation is to understand that most model components are currently imperfect and undergoing evolution. Understanding the fluidity of the environment creates problem resolution contexts where problem resolution occurs more effectively when possible scenarios can be easily tried out and evaluated.
5. *Provide criteria for evaluating the quality of the various solutions.* There are two dimensions to the quality of modeling outputs: structure-based quality and practice-related quality. (These are explained in Chapter 5). With a minimum of understanding, it becomes relatively easy to assess the technical project output quality. In addition, outputs can be assessed for correctness, and end products can be, in essence, certified as accurate representations of the existing system by those working with the system.
6. *Facilitate development of a framework for developing organizational knowledge.* CM² projects are capable of producing a lot of detailed information. It is a non-trivial task to organize and maintain these common metadata to ensure usefulness as inputs to other development efforts. CM² engineering depends on the availability of appropriate data management technologies for project deliverables.

It is crucial to understand that the application of structured techniques is one of the key concepts that makes CM² engineering economically viable. Near-standardized modeling languages and representations also facilitate the process.

In order to achieve optimal or effective use of organizational resources, it is important to match the right person/team with the right task. In most instances, the volume of existing system resources it makes sensible for organizations to groom in-house developers who are able to travel from project to project and become practiced with the technologies and transformations. These developers can be guided using education and practice on small-sized projects. At any level of investment, development of this in-house resource will result in more effective systems maintenance efforts.

The concept of Pareto analysis applies to the transformations.

Eighty percent of the information required by these analyses is contained within twenty percent of the organizational existing systems.

Obviously, a key issue is identifying the 20% of your existing components on which attention should be focused.

4.4.2 Interdependent Metadata Transformations

CM² development is a structured technique (as described above). It is important to understand structured techniques as model development techniques involving model construction and analysis of existing situations and proposed solutions prior to actual system development. The CM² development process provides development teams with tools enabling them to comprehend certain otherwise unfathomable situations.

That is, development process results are interdependent; results of previous transformations influence the form and results of subsequent transformations. In a manner similar to the sequential nature of normalization analysis, each development process transformation is dependent on successful completion of prior phases on a component-by-component basis. Process results are also interdependent; results of previous transformations influence the form and results of subsequent transformations. [another meaning perhaps?] Completion of each phase will yield useful components as shown in Figure 13 below. Presence and absence of these various outputs form the basis for assuring the quality of each transformation.

#	Transformation
Phase I	Understanding the existing environment
T1	Implementation to model
T2	Filtering Technology Out
T3	Planning Integration
T4	Repeatability/Reusability Combining
Phase II	Developing the desired architecture
T5	Potential Capabilities Analysis
T6	Gap Analyses
T7	Solutions Engineering
T8	CM ² -based Component Implementation Engineering

Figure 13 Development process Transformations & Outputs

In Chapters 6 & 7, we describe two sets of transformations that must be applied in order to develop a CM². A total of eight transformations are applied in two phases in order to rapidly develop new components for use in the new architecture. Each transformation uses the outputs of the previous transformation as inputs. Applied in sequence, the transformations are designed to make the CM² more robust, correct, and usefully relevant: a CM² that will contain information that

can be used to effectively support technological aspects of the implementation of business strategy.

The first transformation set is a series of archeology-based, backward-looking analyses required to develop a CM² containing sufficient information to be used as the basis for developing a new architecture. The second set of transformations then incorporates the information developed by the archeology-based analyses and outputs the desired architecture. CM² components are required in order to develop a CM² effectively and efficiently by delivering architectural and business engineering value tied to each rationally sized set of components.

One of the important implications of this transformation interdependence is the required minimization in time that must occur between the archeological transformations and the architectural transformations. Some organizations are tempted to separate these into project phases with waiting periods between phases. The separation of the transformation groups into reporting phases is often useful. However, in practice, it is more realistic to consider them as integrated phases. There are two reasons.

- 1) Delivery of these components requires marginal effort beyond the reverse engineering because of the familiarity of the team with the subject materials. Delivery of these products has often been key to providing direct business value to justifying the project costs.
- 2) Any lag in time between the archeology-based transformations and the architecture-based transformations will see the introduction of changes into the existing environment (changing at a 7% annual rate according to Capers Jones (1999) software productivity research). Management of these changes and integration of the changes into the now out-of-date CM² will require a separate existing system metadata change management system that is not required when the transformations are done in an integrated fashion.

Other reasons include risk of loss of project funding, etc. It is usually better to attempt archeological and architectural-based transformation of some system components and achieve 'transformationally complete' components whose utility can be more easily justified. The alternative process of completing Phase I transformations for a correspondingly larger set of components but not having any of them complete phase II transformations has been more difficult to demonstrate tangible business value to management.

4.4.3 CM² Component-based Development & Versioning

Gone are the days when a data architecture group could say to management, "Wait for two years and we will deliver a wonderful data architecture for you!" Too many large IT project failures have required that management be wary of promises and

implement budget cycle business case analyses. All project work done past the year 2001 will require one-year payoff or it is unlikely that they will be funded.

For this reason alone, it helps to consider CM² development projects as on-going infrastructure maintenance activities that must be re-justified annually. In less than one year time frames, organizations must find the means to implement a “critical mass” of architecture components or they will not be funded the following year. It also means ensuring that management doesn't come to believe that the first year results are sufficient and the project is complete. On-going maintenance will be required to maintain synchronization with typical systems rates of change.

The transformation process can also be described in another manner. Each CM² component can belong to at most one of four versions. The four versions are (in implementation order):

Existing physical -> Existing logical -> Desired logical -> Desired physical

Figure 14 illustrates is the evolution of the CM² components.

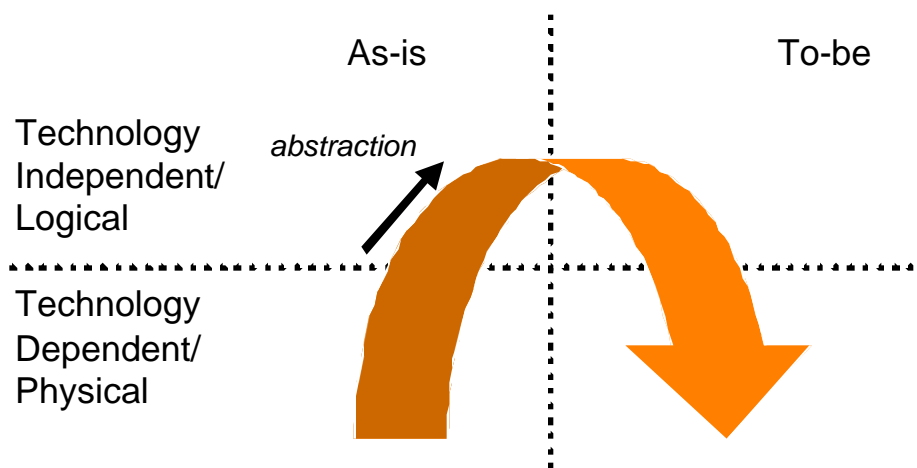


Figure 14 CM² Evolution is technology derived but technology independent

The transformations are performed by CM² development team members and presented for validation to the subject matter experts. The CM² development team will begin by focusing on selectively extracting existing process and technology components and representing them on a one-for-one basis in the **existing physical** portion of the CM². As this is accomplished and component correctness is verified, another activity concentrates on taking the existing physical representations and stripping them of the technological aspects and characteristics: conceptually evolve the components to a **logical existing** representation that is above the technology line. The goal is to focus the model components on what functions the components are supporting, as opposed to how technology provides the support (below the technology line). Only when these transformations are complete can the Phase II transformations be used to prepare the future architecture.

4.4.4 Participation Availability of Subject Matter Expert (SMEs)

The subject matter expert (SME) is typically required to help translate the business systems and business processing so the SMEs and developers have a common understanding. It is very important to respect the time commitments of the SME. Projects have failed due to passive, ineffective, or hostile SME participation. Always recognize that SMEs have full-time responsibilities to satisfy while the developers are attempting to "pick their brains."

Every system has key specialists. For these projects you want what Nardi [1993] calls "gardeners." They are also known as "superusers" and "power users." Regardless of the label, these are users who have developed a better-than-average understanding of the target system. Their expertise surpasses that of the rest of the user community. They are the ones to whom everyone else turns when they encounter system problems.

Key to project success is identifying and obtaining at least available minimal key specialist participation. Because the most effective individuals needed to perform reverse engineering analysis are often the key specialists, DRE projects are potential competitors with other system related projects.

Their principal role is to facilitate the transformation of facts known about the system into knowledge stored as models. For example, the system may have an individual who functions as the primary technical expert, a person who understands the intricacies of the physical implementation of the system, the one to whom everyone turns to answer complex questions. It is just as logical for that person to be a part of the team tasked with developing system enhancements as it is for the same individual to be a part of the reverse engineering analysis team. Because of the cross-functional domain of some DRE projects, project team members can be working on DRE projects in functional areas other than their own. The pressure on organizations to trim staffing levels puts even greater demands on the services of these individuals. It is often critical to identify individual(s) possessing the following technical understanding.

System data specialists. These individuals understand how to get the system to produce the required results. They are the ones other users seek to answer questions such as "What types of transactions do I enter to back out 13 weeks of pay for an employee who has already resigned?"

Policy or regulatory specialists. These include individuals versed in the terminology or lingo of specialized domain areas, such as the Federal Acquisition Regulations or the Americans with Disabilities Act, as well as various federal, state, and regional regulations. They are needed to trace and document the relationships between relevant law, policies, regulations, procedures, and practices, and specific system components implementing the "guidance."

Database specialists. These individuals possess analyst knowledge of the implementation of physical data management for the system. Their knowledge extends to physical representation issues such as whether the data is actually deleted or just marked as deleted.

Application environment specialists. These individuals are cognizant of the program-language (PL) capabilities. Some projects will benefit from the participation of individuals knowledgeable in PL and other systems programming environment characteristics.

Telecommunications specialists. In systems such as the MARS where the distributed nature of the desired system architecture ensures that telecommunications will play an important role, the participation of a telecommunications analyst can facilitate team understanding of relevant related aspects of the system operation. This is required if there are telecommunication aspects, such as electronic data transfer, involved in the system operation.

Operating system specialists. Sometimes systems are implemented in familiar programming languages but on unfamiliar hosts. Operating system specialists can quickly address any specific problem areas or questions. This is required if operating system specific characteristics impact the project. For example, granting access to models in digital form on an unfamiliar operating system often also requires an operating system analyst to help with the extraction.

Portions of this process involve activities such as data integration analyses that require participants to achieve and record common semantic understanding of detailed technical and business information. In addition, a SME will often learn modeling terminology because they recognize the value of the more effective supplement to more typical forms of communication.

The nature of these projects is such that technical and business SMEs are required to achieve mutual understanding to accomplish the desired productivity throughput. These SMEs might include organizational technical and business experts such as DBAs, business experts, data architecture personnel, etc. The team will require access to individuals who understand and can help the team evolve specific SBA components from sources of information such as:

- domain knowledge from the specialists and/or business rules;
- functional descriptions, process models, code, user manuals;
- screen, report, interface specifications, interfaces to other systems;
- logical data models;
- program variables, data element lists, tables, file lay-out structures;
- directives, guidelines, planning statements; and

- program source, object code, job procedures, libraries, directories, test cases, schemas, copylibs, make files, link maps, I/Os and other documentation, data

Our experience indicates that required SME participation will typically be limited to one half-day model refinement/validation (MR/V) session and multiple one-hour question and answer/interview sessions each week for each team that is developing CM² components in parallel. The initial requirements for SME participation will decrease as their knowledge is transferred via the CM² development team into the CM².

A preliminary survey will explicitly consider whether SME participation requirements are typical. The outcome will have an impact on potential SME time commitments. It will be addressed at the preliminary system status review.

CM² components are developed in conjunction with the business oriented team participants. In order to maximize the effectiveness of often very valuable SME time, actual modeling and homework is performed by developers who prepare candidate (often referred to as "straw") model components. These are then introduced to the SMEs in well-structured, JAD-like model refinement and validation (MR/V) sessions.

The goal of the sessions are two-fold and overt:

- 1) obtain corrective feedback and critical improvements to the model components that are developed by the developers and presented to the SMEs; and
- 2) to permit as much involvement on the part of the SMEs as is efficient but also to obtain enough participation in the **refinement** of the model that the SMEs feel ownership of the intellectual property produced.

The latter portion is mandatory: if the SMEs feel that the model is being imposed on them, they will not be able to contribute in satisfactory ways. With satisfactory participation of the SMEs, the organization has the ability to make use of these model components in the same manner as any organizational intellectual property. Organizations that fail to treat these architectural components as intellectual property will be unable to assess the cost of maintaining the components or quantify the benefits associated with component development.

Successful SME participation will be:

- effective – Developers carefully prepare for each interaction with the SME, arriving with agendas and questions in hand.
- efficient – Developer's are responsible for conducting interactions to maximize knowledge transfer within agreed upon boundaries.

- active – SME must be actively involved in development of their respective model components.
- owned – SMEs must conceptually "own" the model components that they develop.
- valuable – SMEs must understand that "this time" the organization is going to treat the intellectual property that is developed as a valuable organizational asset.

4.4.5 *CM² Development Timeline Awareness*

With respect to project timelines, major architectural components can be developed within a single calendar year. These will be used to demonstrate business value – a positive ROI – within a single calendar year. However, it is important that the sponsoring organization understand that it is impossible to completely develop your CM² in a calendar year. The reason is the same that academics refuse to allow certain courses to be taught in fewer than X number of weeks. For example, in the university communities there are often restrictions that say things like, "students cannot learn structured programming techniques in less than 12 weeks time. Similarly, it requires time to assimilate, understand, and fully benefit from the newly developed architectural components. Attempting to fully implement too large a chunk can result in implementation of components based on an immature understanding of the organizational business practices. The question then sifts to: what can be developed to produce enough critical mass that the architectural products and components will be useful to others in the organization? And that they will ask for additional products and components! One strict criterion - that is certain to help avoid analysis paralysis - is that all the facts maintained in the CM² remain officially seen as an organizational project with unfulfilled expectations until the metadata begins to be reused by others within the organization.

4.5 *What is Next?*

Too often the chapter on organizational data quality has wound up at the end of various book sections. We are presenting it up front (so to speak) so that you will understand our commitment to quality as a primary development CM² component. Quality enterprise data is the topic of the entire next chapter. After that Chapter 6 will describe the architecture phase transformations and then chapter 7 will describe the architecture-based transformation.

4.6 *References*

Peter Aiken, et. al. *Reverse Engineering New Systems for Smooth Implementation*. IEEE Software. March/April 1999 16(2):36-43.)

Peter Aiken *Data Reverse Engineering: Slaying the Existing Dragon*. 1996, New York: McGraw-Hill. 394 pages ISBN 0-07-000748-9

Alan L. Eliason, *Systems Development Analysis, Design, and Implementation*, Scott, Foresman and Company, Glenview, Illinois, 1990

Capers Jones *Software Benchmarks and Best Practices* by Caper Jones 1999 Addison-Wesley Professional