A vibrant, abstract background consisting of numerous thin, glowing lines in shades of red, blue, green, and purple. These lines are arranged in complex, overlapping patterns that resemble concentric circles or swirling energy fields against a black background.

WITH VITALSOURCE®  
EBOOK



# GUIDE TO HEALTH INFORMATICS

Third edition

Enrico Coiera



CRC Press  
Taylor & Francis Group

# CHAPTER 9 Information management systems

He will manage the cure best who has foreseen what is to happen from the present state of matters.

*Hippocrates, The Book of Prognostics*

If you can't measure it, you can't manage it.

*Bill Hewlett, paraphrasing an older aphorism of Lord Kelvin*

Not everything that can be counted counts, and not everything that counts can be counted.

*Bruce Cameron, 1963*

Information is care. Every clinical decision is predicated on information about the patient and about the patient's treatment choices. We saw in [Part 1](#) that whenever a decision is important enough, or is made often enough, an information system can be built to support it. In this chapter we examine both how and when healthcare processes and decisions need to be structured from an informational viewpoint. Two key concepts are examined. Firstly, the central notion of an information management cycle is introduced. Second, we explore why it is not always necessary or indeed appropriate to formalize this cycle completely. To do so can be counterproductive, by introducing excessive bureaucracy, especially when flexibility in decision-making is needed. Consequently, we see that many information processes are left in an unstructured or informal state and are more likely to be supported by communication processes.

## 9.1 Information systems are designed to manage activities

The main reason that a clinical information system is built is to manage healthcare activities such as the delivery of therapy or administrative tasks such as deciding staffing levels for a hospital unit. An information model is created to describe the process, including the data elements needed to measure its possible states, and the way that decisions are to then be made based upon those data.

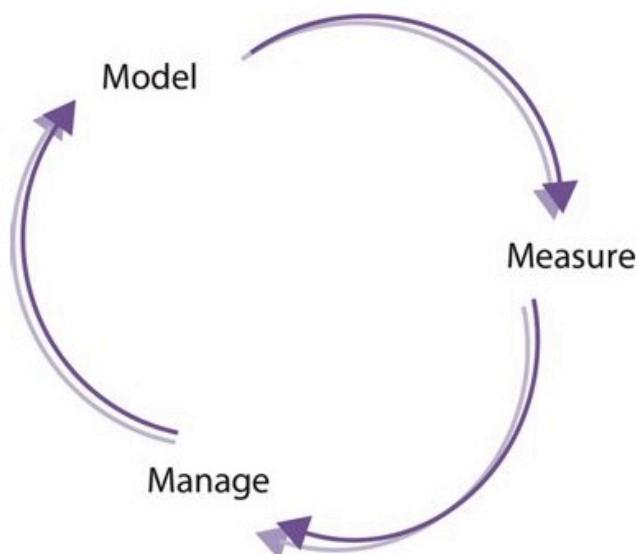
For example, consider the case in which a patient is being treated for an acid–base disorder. The *goal* of management is to maintain the patient's acid–base status in a range consistent with good health. Measurements such as pH and serum bicarbonate provide the data about acid–base state. The model used in this case is physiological. It could include rules that describe the relationships between the measured acid–base parameters in health and in disease. Based upon this model of acid–base disease, one can then interpret the measurement data of a given patient (see [Figure 2.3](#)). Associated with each interpretation, there usually exists a set of *management actions* that can be taken. In this example, the actions are a set of therapeutic interventions.

The process of activity management generally consists of the following steps:

- Define a set of management goals.
- Construct a model of the system that incorporates these goals, as well as which measurements are needed to track performance in achieving goals.
- Measure the system state by gathering those data specified in the model.
- Assess the state of the process being managed by interpreting the measurements using the model.
- If needed, manage the process by taking action to alter its state, and move the process to the target state described in the management goals.

This process is usually iterative. Once an action is taken, a new check usually needs to be made to see whether the process has responded in the right way. When a treatment is given to a patient, measurements are usually taken to see whether that treatment has been effective. If the outcome is not as hoped for, further action may be needed. In this way the result of the first decision feeds a further decision-making round, thereby creating a feedback control loop. Depending on the task, the loop can cycle through many times.

This control loop is the *model-measure-manage cycle*, and it is at the heart of nearly every information system ([Figure 9.1](#)). It is the way in which computer-based information systems are used from the delivery of clinical care to the administration of services and organizations to clinical research. The cycle can be mapped directly onto the model abstraction and application cycle described [Chapter 1](#) (see [Figure 1.2](#)), and it is a simpler way of thinking about how models are used to manage activities.

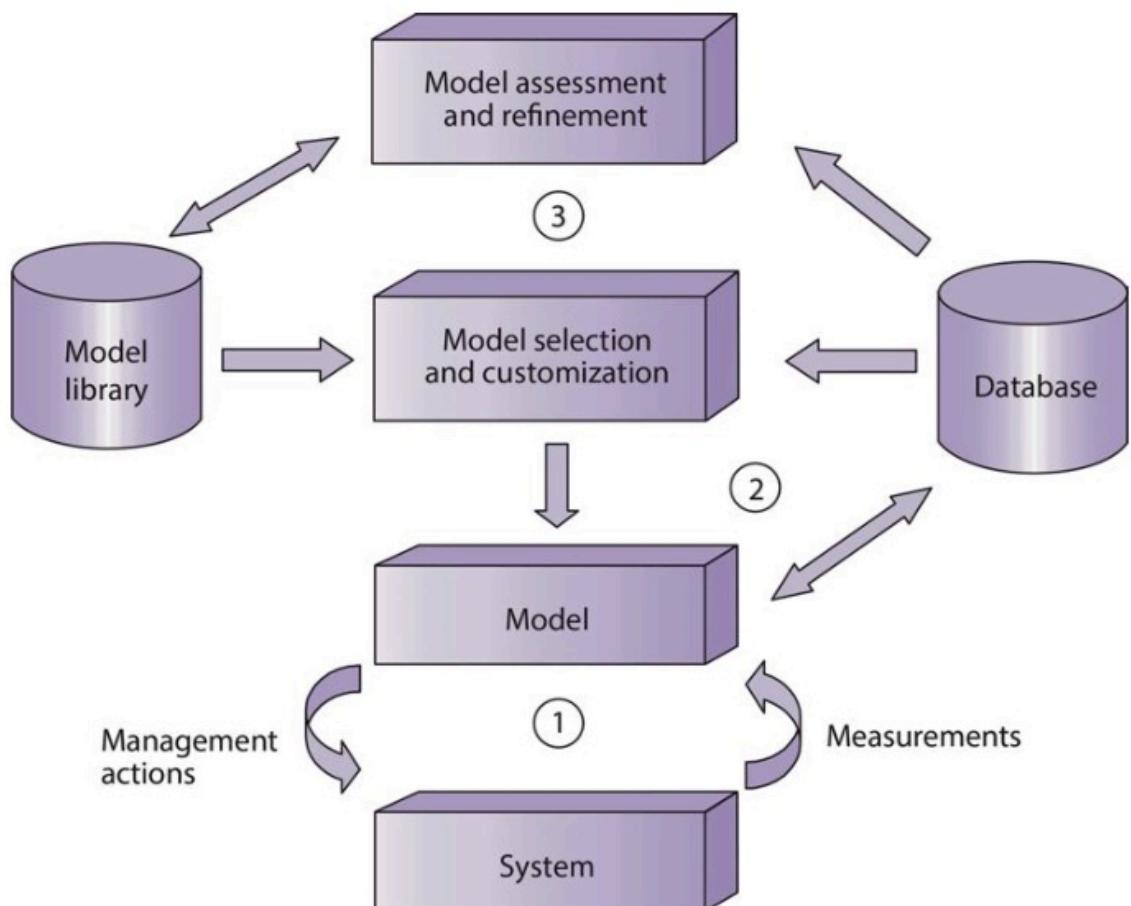


**Figure 9.1** The model-measure-manage cycle. When the outcome of a management decision is fed back into another cycle, a feedback control loop is formed.

There are many alternate ways of describing this process. The *plan-do-study-act* methodology of system control and organizational management (also known as the Shewhart cycle) has been popularized by Deming and is perhaps the most widely known (Deming, [1968](#); Shewhart, [1931](#)).

## 9.2 There are three distinct information management loops

It is useful to distinguish three distinct model-measure-manage cycles. These are responsible for the *application*, *selection* and *refinement* of knowledge. Together, they come together to form the *three-loop model*(Phaal, 1994), which describes the main ways that information is used within an organization, or indeed any system requiring informed control (Figure 9.2).



**Figure 9.2** The three-loop model. Three separate information cycles interact within the health system. The first and second cycles select and apply models to the management of specific systems. The third information cycle tries to assess the effectiveness of these decisions and improve the models accordingly.

We start with a set of models of the system we want to manage. In the case of clinical medicine, these models correspond to medical knowledge. If we are administering an organization such as a hospital, then the models could capture our understanding of economics, historical demands on the service, organizational dynamics and so on. Such models may exist in books and journals, in the software of an information system or in people's heads.

In the first loop, data are used to manage directly a specific activity, such as the selection of a diagnostic test for a patient. The second loop manages the models used in the first loop. For example, a treatment model may need to be replaced or customized more closely to the circumstances of a patient, if sufficient progress has not been made. In the third loop, the performance of all these models is assessed. For example, the clinical effectiveness of a procedure may be measured after its use over a large number of patients, thereby resulting in its modification once problems are identified. Each loop thus has a different role. Each involves different data sources and different operations on the data. As a result, each cycle has different requirements for the kind of information or communication system needed to support it.

## Loop 1 defines the direct application of a model to a task

The first loop is simply the model-measure-manage cycle. Using knowledge contained in a model, measurements are made and actions are taken. For example, a clinician may choose an insulin dosing regimen for a diabetic patient. Having chosen that regimen, the clinician uses it to manage the care of the patient. Measurements of blood sugar levels are taken, and the dose of insulin is varied according to the rules specified in the regimen. Similarly, a hospital has a formula for determining the number of nurses needed to staff a ward adequately. This formula may take into account the number of patients in a ward and their level of dependency. On any given day, based upon measurements of patients' numbers and dependency, a nurse manager will use the formula to decide upon staffing levels for different wards.

## Loop 2 defines the way models are selected and customized

Usually, there are a number of different ways in which a task can be completed – there are many ways that the world can be modelled and many ways in which that model can be translated into a prescription for action. Loop 2 describes the process of deciding which of these models are most appropriate for a specific task.

Returning to the example of the diabetic patient, a clinician must decide which insulin regimen or set of rules is the most appropriate for a given patient. The choice could be based upon an assessment of the severity of disease or on the patient's ability to test his or her own blood sugar and self-administer insulin. The regimen may be varied many times during the period in which the clinician manages the patient. When the patient is ill, the clinician may prescribe 'sick-day' rules that alter the doses of insulin that can be given. If a patient has an implanted insulin delivery device, then the rules for frequency, timing and dose of insulin are different from those for self-administered subcutaneous doses.

In a hospital ward, a nurse manager may need to apply a different formula for staffing levels based upon on the type of ward. A different set of rules may be applied to an intensive care unit, which has different requirements for staff skills and different levels of patient dependency from those of a general surgical ward.

Sometimes local variations need to be considered. Patients have specific circumstances that may require a treatment to be tailored to their specific needs. A patient with particularly 'brittle' diabetes may need to be closely monitored and the treatment varied because of the unpredictability of the disease. Nursing staff on a ward in a teaching hospital may need to vary the balance of their daily activities if they are expected to train students as well as carry out clinical duties. Sometimes these variations in approach reflect very particular circumstances and are not frequent enough to be turned into general policy.

## **Loop 3 is responsible for model creation and refinement based upon the results of application over time**

All knowledge is constantly re-examined, and our understanding of the world evolves with time. In loop 3, the knowledge used to complete a task is examined against the outcome of application. The results of repeated application of a model are pooled, and assessments about its effectiveness are made. When several different approaches to execute a task exist, historical data may help identify the most successful approach. If the models selected in loop 2 are regarded as hypotheses about the best way to approach a task, then loop 1 is where each hypothesis is tested. Loop 3 is where inductive reasoning is applied in the scientific examination of these theories and drives the modification or creation of new theories.

121

Thus, the many different treatment regimens used in the management of diabetes have all hopefully been tested in trials across large numbers of patients. As the outcomes of the treatments are examined, decisions are made about which regimens should be retained and which should be modified and under which conditions particular regimens should be best applied.

Equally, over time, the way in which hospital units are staffed is modified when measurements such as patient outcomes, staff retention and satisfaction levels are examined. Those hospitals that perform best on these measurements can be used as role models by other hospitals that want to improve in a similar fashion.

### **9.3 Formal and informal information systems**

Just because it is possible to define an information system, it does not follow that it is always reasonable to go and build it. If that were the case, our lives would be regulated in minute detail. What happens in reality is that most organizations, small or large, try to find a balance between creating formal processes and allowing individuals to behave freely and informally. Large organizations gain stability through formal processes, but they are often criticized for being overly bureaucratic. Although smaller organizations may be flexible and able to respond rapidly to change, they may be chaotic to work in as a result. The difference between such extremes lies in an organization's view of the need to formalize its internal systems.

So, although structuring processes have clear advantages, including improved reliability, efficiency and consistency, this approach comes with a cost. There are several such costs that need to be considered before a system is formalized:

- First, the creation of a process, by definition and intent, limits flexibility of response. A particular view of how things should be done is captured within the rules, regulations and procedures that describe any formal process. Alternative views exist, and in some circumstances they would produce better results. Choosing to adhere to a single formal process is a trade-off. The number of times it produces a good result is balanced against the cost of those times when an alternate approach would have been better. If the situations being dealt with are highly variable, then it is probably better to come up with a way of handling them from 'first principles', rather than looking to a formal cookbook solution.
- Second, formalizing the information elements that describe a process usually requires considerable effort. An explicit model of the process needs to be created, and this includes definitions of the data that need to be collected. In some situations it is just too costly to engage in this formalization process, given the likely return.

What should now be apparent is that, before a formal information system is created to manage a process, there needs to be an explicit cost benefit analysis. In many circumstances, the result of that analysis may be that it is preferable not to build the system.

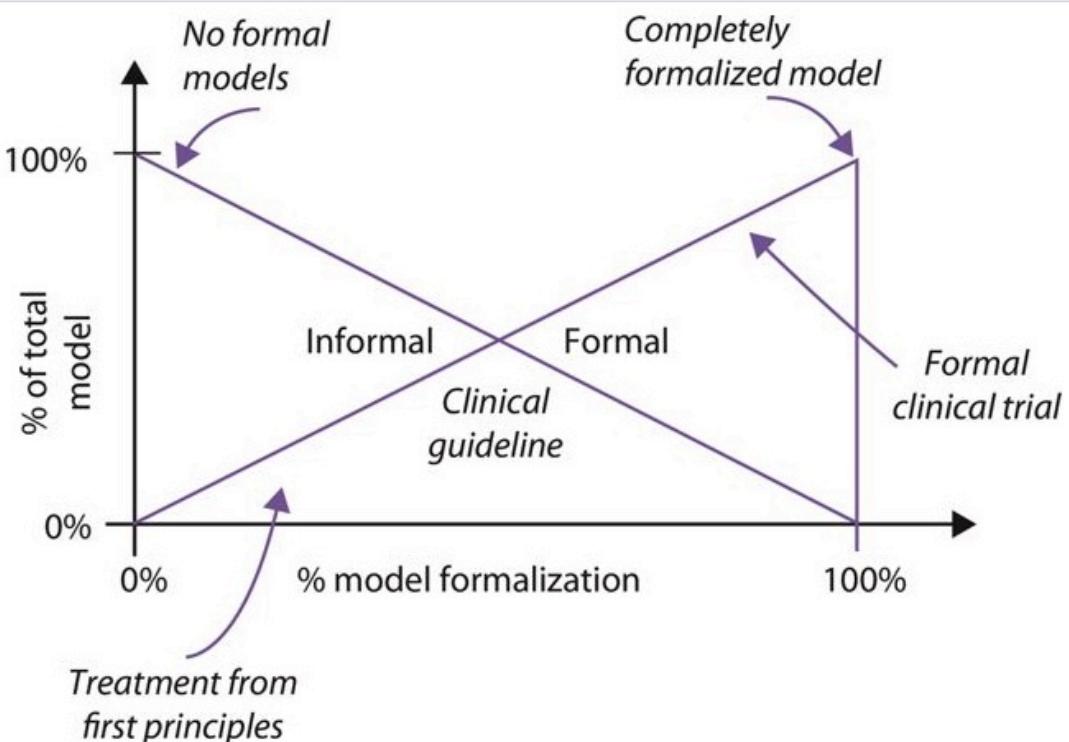


Figure 9.3 There is a continuum of possible formalization of information systems. Formal and informal models combine together to cover the total modelling needs of a given problem. The degree of formalization depends on the type of problem being addressed.

In Chapter 2 we saw that there is a continuum that describes how much of a model is incorporated into a computer and how much is left in the heads of humans (see Figure 2.5). A similar continuum describes those situations in which everything needs to be formalized and those in which a process can be left informal. In between these two extremes, depending upon specific needs, one can formally define some parts of a system and leave the remaining interactions undefined (Figure 9.3). The proportion that is formalized depends on the cost benefits of formalization of a given situation.

Thus, although a hospital has many formally defined procedures for managing different activities, most activities are left informal. Similarly, the way that patients are managed falls along this continuum of formality. Patients enrolled in a clinical trial have their management completely regimented, to maximize the scientific significance of any results. Patients with common conditions may be treated according to well-defined guidelines, but some aspects of their treatment are modified to match individual needs. At the other end of the spectrum, a patient's problems may be approached in a non-standard way, perhaps because of the uniqueness of circumstances. Although there may be a few general guidelines on how to approach such patients, most of what occurs has to be created 'from scratch'. This does not mean that such a treatment, if repeated over time, will not slowly become part of the formal treatment of similar patients.

# Not all data need to be made available for computer interpretation

Given that not all processes need to be formalized, it follows that just because data are stored in a computer system, the data will not always be analyzed. Consider, for example, a recorded voice message left by a member of a healthcare team for a colleague that contains an update on a patient's progress. The message is likely to be loosely structured and cover a number of topics that would not normally be predictable in advance. It is also likely that the message will be of interest to the few individuals associated with the patient's care. There may be no formal agreement in advance on the content of the message or the ways in which it will be used. Such agreements are at the heart of traditional computer-based information systems, in which data formats are specified in advance of the system's construction. Nevertheless, the recorded message constitutes data that are stored, perhaps even on a computer system, and clinical decisions may be altered based upon the content of the message.

122

As a consequence, the message data are treated in a very different way from those data that are formally recorded in a patient's record. The model and the interpretation of the data are owned by the recorder and listener of the message. The machine that stores and transmits the data takes on a passive role. It is used as a data repository or as a channel between the communicating parties, rather than for any active interpretation it may make.

This does not mean that no one would ever want to develop a model that allowed a computer to analyze such a message. Text and speech processing algorithms can be used to do exactly that. When there is a clear cost benefit to doing so, messages may be screened for the appearance of key words or phrases. This may be done for quality control processes or as part of a root cause analysis to understand the circumstances that led to a patient's being harmed.

In general, we can make a distinction between information systems in which data are explicitly modelled and those in which data are intentionally left unmodelled:

1. A *formal Information system* contains an agreed model for the interpretation of data, and data within the system are structured in accordance with that model.
2. An *informal information system* is neutral to the interpretation of data, containing no model and imposing minimal structure on any data to allow it to be stored, retrieved and shared.

Thus, an informal system does not imply the absence of a model or the inability to interpret data in the light of a model. It is just that the model and the act of interpreting the data within an informal system are held somewhere external to that system. In practice, these models are often kept in the heads of those who create or access the informally stored data or in other information systems designed to retrieve and interpret that data using different methods (e.g. text or speech recognition).

In general, informal information systems are used when data are of temporary value, of interest to a very few people, are complex or when the content of the data is not predictable in advance (Coiera, 2000). There is thus no sense in which the data in an informal system are less valuable than formally structured data. Data can be used in fundamentally different ways and in different situations.

123

# Communication systems frequently support informal exchanges

Some tasks, such as storing and retrieving telephone numbers, lend themselves to being formally organized. Other tasks, such as writing quick notes, can be impaired by such structure. People often use simple tools such as pen and paper to manage a variety of unstructured tasks. The key requirements for any tool used to support informal tasks are that it provides a means of creating and then managing information that is flexible and can be used across a variety of tasks.

One of the most common ways of supporting informal processes is to use a communication system to channel data between people because such systems have only minimal models of the content of the data they transfer. Thus, a telephone is an informal information system. The models for interpreting the data transmitted across the telephone are not in the machinery responsible for mediating a conversation, but rather are external to it. When examining the flows of information through an organization, it would thus be a mistake to look only at formal processes because this would give a very skewed picture of what really is going on. There is a complementary, and probably significantly larger, body of information coursing through the informal channels of the organization's communication infrastructure.

123

124

Many processes are a blend of formal information and informal communication elements. Clinical handover is an example of a communication process that benefits from some structure (see [Chapter 5](#)). Structured approaches such as SBAR (situation, background, assessment recommendation; see [Box 5.1](#)) describe both the expected content of the handover conversation, as well as the order in which items are to be presented. This structure stays at a very general level, and it is up to the individuals making the presentation to decide upon the level of detail or issues that need to be highlighted.

The following example should make the contrasting roles of communication and information systems in managing information clearer. Consider a primary care physician who wants to share a patient's electrocardiogram (ECG) with a cardiologist during a teleconsultation. There are two ways in which this could happen:

- **Informal: Scanning the ECG** – An ECG can be captured by a scanner either from a printed version or by sending the ECG document electronically to the scanner. The scanner is not configured in any special way to recognize that it is transmitting an ECG. It just as well could be sending text or a photograph. Thus, the scanner is informal with respect to the content of the data. The cardiologist who reads the scanned image possesses the model for the interpretation of the ECG. The advantage of using a scanner is that it is cheap, easy to use and widely available. Because no interpretive model is associated with the data scanned, the scanner can be used on a wide variety of tasks. The disadvantage of using this system is that it requires someone with expertise on the receiving end of the scanned document to provide the model and interpretation.
- **Formal: Transmitting the ECG data file** – If an ECG is captured directly from an electrocardiograph as a data file specifically structured to store ECGs, then any computer that understands that file structure can take the data and recreate the original waveform. The advantages here are the inverse of scanning an image. Having data that are structured according to a model permits flexible manipulation of the data. One could choose to display only some portions of the signal at different resolutions. The computer could automatically interpret the signal if its models included the different patterns that ECGs may take in disease. The disadvantages of this system are that one requires a dedicated system at both the sending and receiving ends to encode and decode the signal.

The decision to adopt one or other approach is a cost benefit trade-off. A primary care practitioner who rarely has the need to transmit an ECG might find the cost of purchasing a dedicated system unjustifiable. A cardiologist, for whom this is a common occurrence, would find the case easier to make.

In summary, an informal information system can be used on many different tasks, but it performs each of them less effectively than a system formally designed for the task. When tasks are infrequent, it is often more cost-effective to use an informal solution. In contrast, as a task starts to become more frequent or important, then more expensive and specific formal tools can be created to support it.

124

125

## Discussion points

1. Are there any significant differences between the model-measure-manage cycle and the plan-do-study-act cycle?
2. Pick a specific health service or clinical task that you are familiar with, and describe its function in terms of the three-loop model.
3. Is the model, measure and manage cycle a positive or a negative feedback system?
4. When is it not appropriate to introduce a computerized information system into an organization?
5. Describe the ways in which you could design a diabetes information service for consumers, first by using a formal information system approach and then by using only communication technologies.
6. Some clinicians seem to prefer using communication devices such as mobile phones, or speech-driven devices such as dictation recording apparatus, rather than computers. Why may that be?
7. What is the appropriate mix of formal and informal processes at patient handover?

## Chapter summary

1. An information system is developed to manage a set of activities, and its function can be characterized as repeated cycles of modelling, measurement and management.
2. There are three separate information loops. Loop 1 defines the direct application of a model to a task. Loop 2 defines the way models are selected and are customized. Loop 3 is responsible for model creation and refinement based upon the results of application over time.
3. There are considerable advantages to structuring information processes, including improved reliability, efficiency and consistency. There are also costs associated with formalization, including lack of flexibility to accommodate varying circumstances and the effort involved in defining the system.
4. Not all data need to be available for computer interpretation. A trade-off exists between creating explicit models that permit formal information systems to be created and leaving the system in an informal state, with minimally defined models and data.
5. A formal information system contains an agreed model for the interpretation of data, and data within the system are structured in accordance with that model.
6. An informal information system is neutral to the interpretation of data, contains no model and thus imposes minimal structure on any data that are contained within the system. The model and interpretation for data contained within an informal system are external to that system.
7. Communication systems are frequently used to support informal exchanges. Information flow through an organization occurs both by formal processes and through the informal channels of the organization's communication infrastructure.
8. Informal information systems are used when data are of temporary value, of interest to very few people or are complex or when the content is not predictable in advance. When tasks are infrequent, it is more cost-effective to use an informal solution. However, if a task requires formalization because of its frequency or importance, more expensive and specific tools can be brought in to support it.

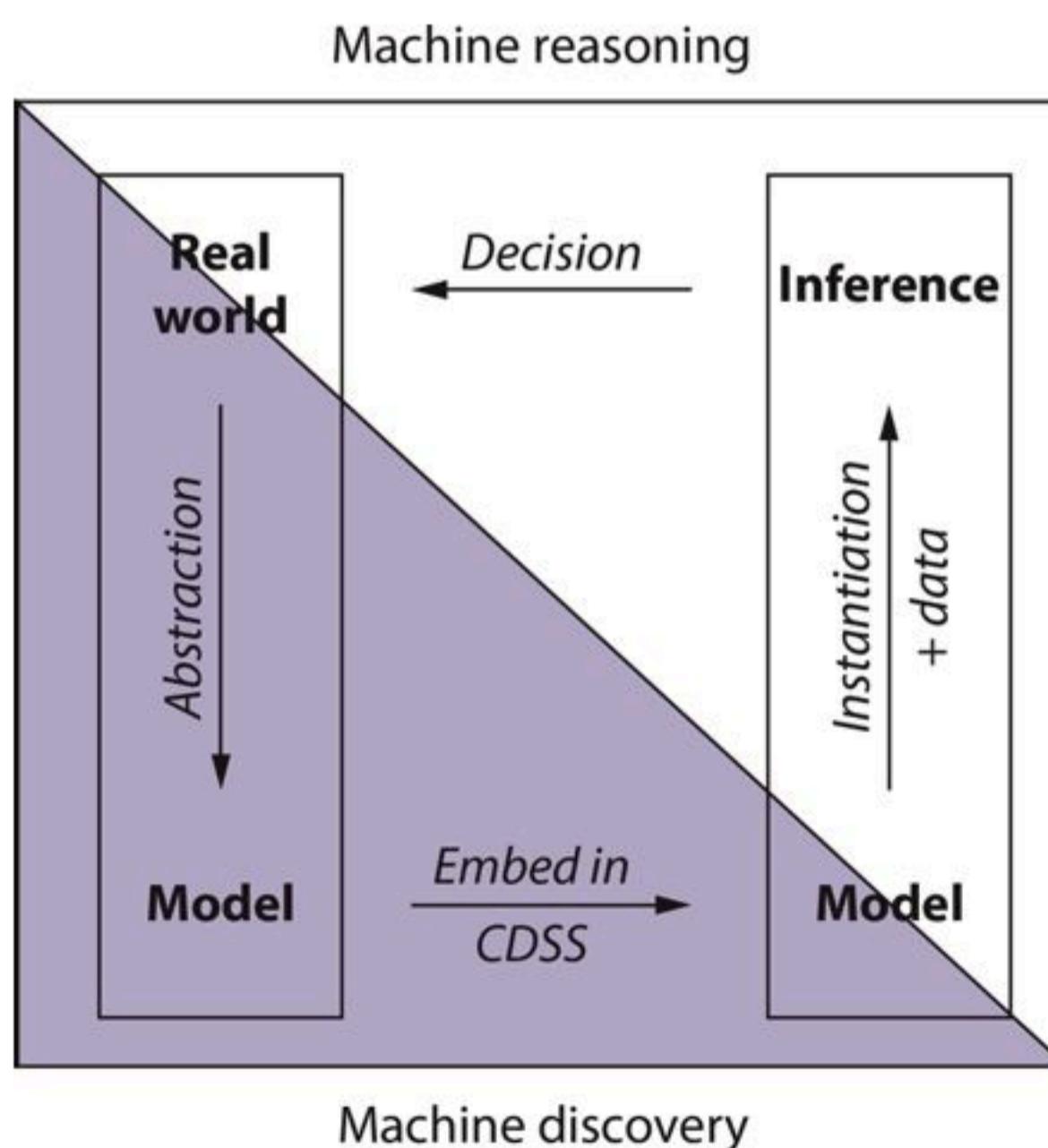
## CHAPTER 27 Model building for decision support, data analysis and scientific discovery

### 27.1 Before reasoning about the world, knowledge must be captured and represented

In [Chapter 26](#), we were introduced to some of the different ways in which knowledge can be represented computationally and then used to draw automated inferences. In this chapter we explore the origins of that knowledge. In some situations, the knowledge embedded in a clinical decision support system (CDSS) is a direct translation of well-understood principles captured, for example, in guidelines or textbooks – the CDSS essentially replicates what is already known. In some circumstances, however, little may be known, or there is significant variation in system behaviour from one setting to another that has not been accounted for. In such circumstances, we use a different class of computer methods to assist in discovering new relationships in the world.

These *computational discovery* methods typically take as their input data from the process that needs to be understood, and they then output knowledge which may be in any of the forms we encountered in [Chapter 26](#) – rules, networks, models. This encoded knowledge can then be placed into a CDSS capable of taking similar data as input and applying the knowledge to make a decision. Together, machine reasoning methods such as CDSSs and machine discovery systems complete the model cycle introduced in [Chapter 1](#) from model construction through to use ([Figure 27.1](#)).

In this chapter, we review the different applications of machine discovery before exploring the main methods used to acquire knowledge, whether for use in a CDSS, to support the analysis of new data or to drive scientific discovery. The chapter also covers how one evaluates the correctness of these models and presents an approach to data analytics, based on machine discovery methods.



**Figure 27.1** Computer reasoning depends first on acquiring models that record our understanding of how the world works and then applying those models in a decision support system to arrive at conclusions shaped by the data specific to a given decision task.

## 27.2 Computational discovery systems find wide application in healthcare

Computational discovery systems have three broad uses. The first is to create models that can then be used within a CDSS – a process sometimes called knowledge acquisition. Next, these systems can be used as an adjunct in the scientific discovery process, helping scientists understand the meaning of data. As data sets grow in size, machines start to become essential in identifying promising hypotheses to explain data patterns. The third application area is data analytics, in which we try to discover meaningful regularities in data, much as scientists do, but with the goal of using this information directly to help in managing a process such as running a hospital or health system. Where scientists engaging in discovery would treat data mining discoveries as hypotheses to be tested in new experiments, analysts would use their discoveries to intervene directly in the system they are trying to manage.

### Association rule mining

Discovery systems can be used to develop the knowledge bases for a CDSS. Given a set of similar clinical cases that act as examples, a discovery system can try to identify which clinical features among the cases are most uniquely associated with a diagnosis, often expressed as rules. A classic example of this type of discovery system is KARDIO, which was developed to build and then interpret electrocardiograms (ECGs) (Bratko *et al.*, 1990). Association rules can be mined from historical electronic health records (EHRs), for example, to link medications with the clinical problems they most commonly treat. When new patients are admitted to hospital, the rules can use a patient's medication list to automatically generate the corresponding problem list (Wright *et al.*, 2013). Other discovery systems learn decision trees instead of rules (Quinlan, 1986). These trees look like the decision trees discussed in [Chapter 8](#) and can be used to guide diagnosis or treatment decisions.

A similar approach can be used when *data mining* for previously unknown relationships hidden within large databases. For example, EHRs contain huge quantities of data, and buried with these records potentially lies much new knowledge (Jensen *et al.*, 2012). It may be that patients treated with a drug for one disease also gain benefit from the drug for another disease they have. This additional application of the drug may not be known, but by aggregating and comparing different patient groups, it should be possible to uncover the previously unexpected benefit. One can look for such associations in EHRs by searching for patients with a disease, looking within that group for patients who had a better prognosis than average, and seeing whether this group is associated with the prescription of medications for other diseases.

Previously unreported side effects of a drug can also be discovered by *text mining* adverse event reports or by looking for higher than expected rates of adverse events among patients taking a drug, compared with similar patients taking a different drug (Tatonetti *et al.*, 2011). Such text mining has application anywhere that we have assembled large databases of relatively unstructured text. One can look for patterns in the text of medical notes, including laboratory or imaging test reports. Adverse event reports, which are generated when patients are potentially or actually harmed during the process of care, are another rich source of information, and text mining can be used to discover important patterns in such data, as well as to generate CDSSs that automatically flag when a new report is serious and requires immediate attention (Ong *et al.*, 2012).

## Process mining

It is often the case that there are several alternate treatments for a given condition, with slightly different outcomes. It may not be clear, however, which features of one particular treatment method are responsible for such variation in outcomes. Computational discovery methods have a role in the development of treatment guidelines, which essentially describe the most suitable treatment process for a patient (Abston *et al.*, 1997; Toussi *et al.*, 2009). *Process mining* is the general term for discovery techniques that take temporal sequences of events and use them to build models of common sequences. For example, extracting sequences of events captured in an EHR can be used to describe patient flows, such as the care pathway for patients who are admitted to a hospital and then move through a stroke unit or oncology ward (e.g. Huang *et al.*, 2013; Mans *et al.*, 2008; Mans *et al.*, 2009).

Process models may be represented by simple event sequences or more complex networks. When there is a pre-existing process description, such as a treatment guideline, then process mining methods can be used to look for how frequently the ideal process is executed and uncover common exceptions to the process. This information may suggest either the need to make local processes conform more closely to the guideline or to make changes to the guideline if the exceptions are well motivated (e.g. Mani *et al.*, 2007). Building process models can help explore the reasons behind variations in the execution of a specific procedure in the hands of different surgeons (Forestier *et al.*, 2012) or interventional radiographical studies such as angiography (Gentric *et al.*, 2013) and can contribute to skills assessment and targeted training.

456

456

## Literature-based discovery

Discovery methods can be applied directly to the research literature, to uncover previously unsuspected associations. The Arrowsmith system is a classic literature-based discovery system, allowing a user to ask questions such as, 'Are there any things that are shared by patients with disease A as well as disease C?' Such questions allow insights about the management of one condition to be re-applied to the associated condition. In 1986, Don Swanson used Arrowsmith to connect papers about dietary fish oil with papers about Raynaud's syndrome. Some papers described how dietary fish oil reduced blood lipids, platelet aggregability, blood viscosity and vascular reactivity. Papers about Raynaud's syndrome noted that it was a peripheral circulatory disorder associated with and exacerbated by high platelet aggregability, high blood viscosity and vasoconstriction (Swanson, 1986). Swanson hypothesized that fish oil would thus be beneficial in the prevention and treatment of Raynaud's syndrome, a hypothesis that has since been tested and validated.

## Biological model discovery

Beyond such associational discoveries, it is possible to use patient data to construct pathophysiological models that describe the functional relationships between various measurements. For example, a learning system can take real-time patient data obtained during cardiac bypass surgery and then create models of normal and abnormal cardiac physiology (Hau and Coiera, 1997). These models may be used to look for changes in a patient's condition if they are applied at the time they are created. Alternatively, if used in a research setting, these models can serve as initial hypotheses that can drive further experimentation. Model discovery finds great use in hypothesizing new genetic regulatory mechanisms, metabolic pathways and disease processes.

## Drug discovery

One particularly exciting application of learning systems is to discover new drugs. The learning system is given examples of one or more drugs that weakly exhibit a particular activity, and based upon a description of the chemical structure of those compounds, the learning system suggests which of the chemical attributes are necessary for that pharmacological activity. Based upon the new characterization of chemical structure produced by the learning system, drug designers can try to design a new compound that has these characteristics. Traditionally, drug designers synthesized a number of analogues of the drug they wished to improve upon and experimented with these analogues to determine which exhibited the desired activity. By using discovery tools, the development of new drugs can be speeded up and the costs significantly reduced. Statistical analyses of drug activity have been used to assist with drug analogue development (exploring different molecular variations of a drug), and computational discovery techniques have been shown at least to equal if not outperform chemists, as well as having the benefit of generating knowledge in a form that is easily understood by them (King *et al.*, 1992).

456

457

## 27.3 Manual knowledge acquisition methods can guide interactions with human experts to reveal the basis for their judgements

The first CDSSs used manually developed knowledge bases, typically obtained by describing the reasoning used in the heuristic judgements of experts. These rules described the features associated with different diagnoses and were used to generate a differential diagnoses or suggest the most appropriate therapy for a clinical case. The methods used to craft these rules ranged from the highly informal to the systematic use of robust processes designed to elicit the rationale underpinning human judgements. Despite their often qualitative nature, these ‘manual’ methods remain an important model building tool, both used alone and to help guide automated processes. In the latter situation, although we may have huge data sets with very many data types, the process of automated model building often needs direction from those who understand the problem domain – for example, identifying which patterns are interesting or which data types are most likely to be significant in shaping an outcome.

There are several classic approaches to *knowledge acquisition* or *knowledge engineering*, and these are very similar to qualitative methods widely used by researchers when they seek to answer a variety of other questions (Cooke, 1999; Olson and Rueter, 1987; Welbank, 1990):

### Interviews

Especially early in the process of understanding a decision task, interviews can be very illuminating. Both unstructured (free flowing) and structured (following a question script or describing a typical scenario) interview methods can be used. Structuring interviews is more appropriate when there are clear questions to be answered, and unstructured approaches are more useful in the very early sense-making stages of the process. Interview data can be collected on anything from note form through to full video recording.

## Observation

By observing humans as they make judgements, either in real world settings or, for difficult situations, in response to artificial cases, we can develop an understanding of the types of decision tasks they face and how they resolve them. Observers can take on active, non-participatory or highly passive roles and capture what they observe through notes, audio, images or video. Many of the methods used to analyze data are borrowed from anthropology and ethnography. These include *grounded theory* (Strauss and Corbin, 1994), in which researchers gather data in as neutral a way as possible and then seek the concepts implied by the data by constant comparison across data types, supported by additional data gathering. Researchers take note of how frequently new concepts emerge from the data analysis and seek the point of *saturation* (when no further new concepts are evoked) to indicate that the knowledge acquisition task is likely complete. Advantages of observation are the richness of the data that can be captured and their reflection of reality, as opposed to recollection. Disadvantages include the difficulty in capturing rare events or understanding complex processes by observation alone.

457

458

## Process tracing

Procedural descriptions can be obtained by observing a sequence of behaviours required to carry out a particular task. *Think-aloud protocols*, for example, ask subjects to explain what they are thinking as they carry out a predetermined task, such as operate a device, use a piece of software or engage in a clinical process. In addition to capturing what is said and logging event sequences, subjects can also have a variety of other measurements taken, including eye tracking to see where they focus at any particular moment and physiological stress measurements if the setting is one in which operator stress is an issue. Some tasks may be so onerous that it is not possible to explain what is happening from moment to moment, and in these cases subjects are asked to recount what was happening later, perhaps in response to recorded data such as video. Process tracing methods generate potentially very rich data sets that can require substantial effort to analyze. Computational tools can assist in process analysis, for example, by looking at process traces from a number of different subjects and across different example cases and then generating typical pathways based upon statistical analyses. Such methods also find use in designing user interfaces for software because they reveal the paths that users take as they try to operate software, thus informing designers of the difficulties users face when confronted with their systems.

## Conceptual methods

In order to understand a decision task completely, it is necessary to describe all the underlying concepts in the domain and their relationships. If we were eliciting knowledge about the diagnosis of a disease group, then the concepts would include the diseases that are under consideration, the clinical symptoms and signs that a clinician may uncover and the different tests available and their typical results for different diseases. Conceptual methods are formal approaches to capturing these underlying structures, and they tease apart the ontology and conceptual basis of a domain, as well as the knowledge about these relationships. Methods can assist in inducing subjects to explain whether specific concepts are related and to what degree. The *repertory grid* method asks subjects to rate the relatedness of concepts along highly structured dimensions. For example, when attempting to relate a clinical sign to a different disease, clinicians may be asked to make that judgement according to expected frequency, ease or cost of elicitation and whether the sign is pathognomonic.

## Case-driven knowledge acquisition

Knowledge bases typically need to be updated as knowledge changes or as the role of a CDSS expands. As new rules or other elements are added, they can introduce inconsistencies, with new elements overlapping or even contradicting each other, a result in part because ‘many hands’ are involved. To circumvent this knowledge maintenance problem, computational tools can carefully guide the process of adding knowledge to a system. In the Ripple-Down Rules (RDR) methodology, experts start with an essentially empty CDSS and add new rules for each new case encountered that the CDSS cannot handle (Compton and Jansen, 1990). The new knowledge that ‘repairs’ the system’s performance on a case is attached to the portion of the classification tree that failed. To ensure that the changes do not introduce errors, a set of cornerstone cases is run against the updated CDSS, containing the data describing each case and the expected ‘gold standard’ response to the case. Despite a process that is driven by new cases, rather than by abstract principles of form, and that seems to produce quite large and messy knowledge bases, CDSSs built using RDR appear much easier to build and maintain and to perform just as well (Compton *et al.*, 1992). Many commercial systems are based on RDR, for example, to assist with bulk interpretation of high-volume laboratory results (Compton *et al.*, 2006). Although initially developed to manage single classification tasks in pathology, RDR methods have evolved to support multiple classification tasks and have found application across a wide range of domains and problem types including configuration, simulation, planning and natural language processing (Richards, 2009b).

458

459

## Crowdsourcing

Instead of relying on individual interviews or observations and leaving knowledge synthesis to the *knowledge engineer*, users and experts can engage collaboratively in the development of a knowledge base. Crowdsourcing can be used with great accuracy to identify key concepts and relationships between concepts in a domain, such as the different drugs typically prescribed for given clinical problems (McCoy *et al.*, 2012). Clinicians can also be asked to agree on their recommendations for the best decision in different cases, and these decisions then can be used as a gold standard against which to test a CDSS (Wagholarikar *et al.*, 2013). Wiki-style collaborative tools have also been developed to support case-based RDR knowledge acquisition (Richards, 2009a). The value of the crowdsourced approach is that many different individuals can, in a controlled way, suggest key concepts and knowledge elements. These can then be communally critiqued, and new evidence can be brought in to correct misconceptions, all in a way that a generic knowledge engineer, unfamiliar with a specific domain, is unable to do.

## Systematic review

All these methods presume that ‘experts’ working in a domain are the main sources of knowledge that is to be codified. It is often the case, however, that the decision task for a CDSS is well described in the scientific literature. As we saw at the end of Chapter 7, a formal process such as systematic review can be used to search for and select research studies into the efficacy of different treatments for a disease and to then synthesize the evidence so that a recommendation can be made about the best course of treatment in different circumstances. The resulting clinical guidelines are documents can then be translated in a computer representation such as rules or a protocol.

## 27.4 Computational discovery systems generate new knowledge from data

All scientists are familiar with the standard approach to data analysis. Given a particular hypothesis,  
statistical tests are carried out to see whether the hypothesized relationship can be found among specified  
variables in a data set. *Computational discovery systems* typically hypothesize such relationships among  
different variables on their own and then test to see whether these fit a data set. The methods for testing  
relationships include classic statistical methods, information theory and *machine learning*.  
459  
460

### Supervised and unsupervised learning

The learning task for a machine can take two major forms. In many cases, a data set will come with some class labels attached to the data. For example, a set of patient records (the data) is categorized as normal or diabetic (the class labels). The discovery system then sets about trying to understand which attributes of the data set are uniquely associated with each class. In contrast, when data are unlabelled, the learning task is more complex because the machine needs to hypothesize its own classes.

*Supervised learning* algorithms are designed to learn from labelled examples. We provide the algorithm with examples of two or more classes (or labels) and ask it to find a model that best distinguishes between the classes. For example, a supervised algorithm can be tasked to distinguish patients who have a low risk of breast cancer from those with the disease, by using their medical records. A set of *positive examples* – patients with breast cancer – is identified manually from the medical record, typically by experts, along with a set of *negative examples* – patients who do not have the disease and appear at low risk. Each example is a data record built out of different data elements (e.g. blood pressure, age, biomarkers, expressed genes in a tissue), and these individual variables are called *features*.

The learning algorithm is then shown a portion of the data (including an equal number of positive and negative examples) as its *training set*. Using these data, the algorithm builds a model designed to best separate the classes, based upon the values of the features. For example, the algorithm might identify that possessing the gene *BRCA1* seems to be common to patients with breast cancer, but not the negative examples. To validate the model built by the algorithm, the remaining unseen portion of the data is used as a *test set*. The model developed in the training phase is applied to the data as if it was a functioning CDSS, and the machine-predicted class is compared with the true label. Once classification performance is acceptable, the learned model can be used in a CDSS.

In contrast, *unsupervised learning* algorithms have access only to an unlabelled training set. The purpose of these algorithms is to ‘data mine’, in which they try to discover new classes within the data, as well as to provide the descriptions of each class. For example, an unsupervised algorithm may look at the EHR data from patients with breast cancer and try to discover sub-groups of these patients with different disease presentations or treatment responses. The algorithm may, for example, discover that there is a group of patients with *BRCA1* and *BRCA2* genes whose prognosis and response to treatment are clearly different from those of other patients with breast cancer. The algorithm will try to identify the features that distinguish the sub-groups, and this information could then guide scientists to examine what biological or other explanation exists for this previously unseen difference.  
460  
461

When class labels are clear or the cost of obtaining the labels is justifiable, then a supervised learning approach makes sense. When labelling is difficult or impossible, unsupervised methods are the only possible approach.