

Chapter 2

Basic Concepts and Terms



2.1 Introduction

Health informatics specialists usually deal with ambiguous terms based in computer science, medicine, health sciences, business informatics, and related disciplines. In practice, ambiguous terms lead to difficulties and errors in communication.

One major objective of this textbook is to provide the reader with a clear *terminology*, i.e., a system of concepts and related terms, for health information systems and their management. Such a terminology helps health informatics students, practitioners, and scientists in specifying, describing, and communicating their objectives, tasks, and working results (Fig. 2.1).

This chapter introduces the terminology for health information systems and their management as used in this book. For describing information system architectures for health, we introduce the three-layer graph-based metamodel. It links the logical and physical tools that are used in health information systems to health care functions, which describe the tasks to be performed in certain health care settings, for example, patient admission or execution of diagnostic procedures in a hospital.

To support the learning of the terminology provided in this textbook, some of the authors have developed an ontology called SNIK (semantic network of information management in hospitals) that contains the most important concepts and terms of this book together with their relations to each other.¹ In addition, the ontology links our terminology to other terminologies from business informatics and management of information systems. This gives the reader a holistic view of the management of information systems in health and its overlap with other disciplines.

After reading this chapter, you should be able to

- explain the difference between data, information, and knowledge,
- define (health) information systems and their components,

¹<https://www.sn timer.eu>.

Metadata is data about data. Metadata provides information about one or more aspects of data such as the purpose of the data, author and time of creation, used standards, or file size.

Data cannot be interpreted by a person without knowledge about the documentation context. To be reinterpretable, there must be an agreement on how data represent information.

After the physician found the note saying “Russo”, “8.5”, and “++”, she meets a nurse who tells her that the note documents the fasting blood sugar level of the patient Jakub Russo. Now the physician can interpret the note. “Jakub Russo has a fasting blood sugar level of 8.5 mmol/L” is health-related information about the patient Jakub Russo.

There is no unique definition of information. Depending on the point of view, the definition may deal with a syntactic aspect (the structure), a semantic aspect (the meaning), or a pragmatic aspect (the intention or goal of information). We want to define information as follows:

Information is a context-specific fact about entities such as events, things, persons, processes, ideas, or concepts. Information is represented by data.

What does the symbol “++” on the note mean? The physician can interpret this data because she has knowledge about blood sugar levels. She knows that fasting blood sugar levels below 5.5 mmol/L are normal, from 5.6 to 6.9 mmol/L are an indicator for prediabetes, and above 7.0 are an indicator of diabetes.

Knowledge is general information about concepts in a certain (scientific or professional) domain (e.g., knowledge about diseases or therapeutic methods) at a certain time.

Knowledge as general information contrasts with specific information about particular individuals of the domain (e.g., information about a patient). This means that, due to the physician’s general knowledge about diabetes symptoms, she can conclude that Jakub Russo suffers from diabetes, which, in turn, is information about Jakub Russo.

Although a paper note saying “Russo”, “8.5”, and “++”, and its subsequent interpretation, is not an example of systematic data, information, and knowledge

processing in health care, it may be helpful to understand the difference between data, information, and knowledge.

However, in the context of health information systems and beyond, it is sometimes difficult to distinguish between the processing of data and information. Does an *application system* for patient administration process data or information during *patient admission*? Do physicians process data or information when they make a diagnosis? Throughout this book, we use the terms “data,” “information,” and “knowledge” as precisely as possible and want to emphasize the differences between them. Therefore, the reader should be aware that we have given careful thought to the use of terms containing “data,” “information,” and “knowledge.”

2.3 Health care Settings

In accordance with the World Health Organization (WHO) [1], we regard settings as places, social contexts, or facilities where people actively use and shape the environment and thus create or solve problems. In these settings, life situations take place. Within these life situations, creating and solving problems requires and causes complex *information processing*.

If people actively use and shape the environment and thus create or solve problems in settings related to health care, we call these settings *health care settings*. Cities, villages, private homes, medical offices, hospitals, health care regions, *health care facilities*, and *health care networks* are all health care settings. And again, solving problems related to health care is essentially characterized by intensive information processing.

2.4 Systems and Subsystems

Before considering the details of information systems, let us first define the concept of a system.

A *system* is a set of persons, things, events, and their relationships forming an integrated whole.

We distinguish between natural systems and artificial (human-made) systems. For example, the nervous system is a typical natural system, consisting of neurons and their relationships. An artificial system is, for example, a hospital, consisting of staff, patients, relatives, and their interactions. If a (human-made) system consists of both human and technical components, it can be called a *socio-technical system*.

A system can be divided into *subsystems* that comprise a subset of the components and the relationships between them. For example, the sympathetic nervous system is a subsystem of the nervous system. A subsystem of a hospital is, for example, a ward with its staff and patients. Subsystems themselves are again systems.

For professionals, however, the term “system” is often not specific enough and needs to be refined in order to avoid misunderstandings (“If you don’t know what it is, call it a system.”).

2.5 Information Systems

Focusing on processing, storing, and providing data, information, and knowledge in settings leads to the term “information system.”

An *information system* is defined as that socio-technical subsystem of a setting which comprises all data, information, and knowledge processing as well as the associated human or technical actors in their respective data, information, and knowledge processing roles.

As stated above, if a (human-made) system consists of both human and technical components, it can be called a *socio-technical system*. But what does “socio-technical” mean when looking at the information system of a given setting? “Socio” refers to the people involved in data and information processing (e.g., health care professionals, patients, medical or health informaticians), whereas “technical” refers to tools such as computers, software, telephones, and paper-based patient records. Thus, when considering the information system of a setting, the people and tools in this setting are considered only in their role as information processors, carrying out specific actions following established rules. A physician carrying out *medical admissions* of patients in a hospital, for example, follows established rules for interviewing and examining the patients and documenting their answers in *medical documentation and management systems* (MDMS).

An information system can be divided into subsystems called *sub-information systems*. For example, the information system of a setting can typically be split into two sub-information systems: the part where computer-based tools are used is called the computer-based sub-information system; the rest is called the non-computer-based sub-information system of the information system. Information systems of health care settings may also be divided by organizational structures (e.g., sub-information system of surgical departments or sub-information system of departments for internal medicine) or by professions (e.g., sub-information system of nursing and sub-information system of medical treatment).

2.6 Health Information Systems

Health information systems support health care professionals working in health care facilities as well as healthy or sick persons in their different life situations. The life situations as introduced in Sect. 1.2 are linked to various health care settings, such as health care facilities, where *prevention*, *patient care*, or rehabilitation are carried out. Such situations are also linked to the personal home environment, where people care for their own health or for the health of their relatives and where they solve health-related problems.

Obviously, health care cannot be considered an isolated procedure taking place in one *health care facility* (e.g., one hospital or one medical office). Instead, health care is a patient-oriented process encompassing *prevention*, diagnosis, and therapy going beyond the facilities' boundaries and integrating the home environment. The patient-oriented care process thus takes place in networks of different actors. Such actors are, for example, hospitals, medical offices of general practitioners (GPs), pharmacies, rehabilitation centers, home care organizations, and even health insurance companies and governmental authorities. We call these networks health care networks. Health care networks can also be understood as health care settings.

With the definition of information systems in mind, a health information system can thus be easily defined:

A health information system (HIS) is the socio-technical subsystem of a health-related setting which comprises all data, information, and knowledge processing as well as the associated human or technical actors in their respective data, information, and knowledge processing roles.

A health information system that uses computer-based data processing and communication tools is called a *computer-based health information system*. Please note that health information systems typically comprise both computer-based as well as non-computer-based sub-information systems.

If we refer to the information system of a certain health care facility, such as a hospital or a medical office, we can use the more specific terms “hospital information system” or “medical office information system,” respectively. The information system of a health care network can be called a *transinstitutional health information system (tHIS)*.

As a consequence of this definition of a health information system, a health care setting has a health information system from the beginning of its existence. Therefore, the question is not whether a health care setting should be equipped with a health information system, but rather how its performance can be enhanced, for example, by systematically managing it or by introducing state-of-the-art tools.

We will describe health information systems in more detail later in this book, especially in Chaps. 3 and 6. In Chap. 3, we will discuss general characteristics of

health information systems. In Chap. 6, we will discuss special characteristics that arise for information systems in specific (health care) settings.

2.7 Information Logistics in Health Information Systems

The goal of a health information system is to sufficiently enable *patient care*, administration, and management. For some types of health care facilities, such as university medical centers, health information systems also have to enable research and teaching.

While managing health information systems, legal and other requirements must be taken into account. Legal requirements encompass, for example, data protection or reimbursement aspects. Other requirements may result from management decisions, such as building a common EHR in a transinstitutional information system.

To sufficiently enable *patient care*, administration, and management, health information systems must do the following:

- It must make information, primarily about patients, available. Current information should be provided on time, at the right location, to authorized staff, and in an appropriate and usable form. For this purpose, data must be correctly collected, stored, processed, and systematically documented in order to ensure that correct, pertinent, and up-to-date patient information can be supplied, for instance, to physicians or nurses, so that they can make the right decisions.
- It must make knowledge available, for example, about diseases, side effects, and interactions of medications, to support *decision-making* in diagnostics and therapy.
- It must make information available about the quality of *patient care* and the performance and cost situation within the health care setting.

We can summarize this under the term *information and knowledge logistics*.

Information and knowledge logistics aims at making the right information and knowledge available at the right time, at the right place, to the right people, and in the right form so that these people can make the right decisions.

So what is meant by the “right place”? Persons responsible for information and knowledge logistics in health information systems must consider various areas of health care settings. Health care networks, for example, can consist of

- hospitals,
- medical offices,
- rehabilitation centers,
- nursing homes or ambulatory nursing organizations, and
- personal environments, especially patients’ homes.

Who are the “right people” to be provided with the “right information and knowledge”? Obviously, the most important people in a health care setting are the patients and, in a certain respect, their informal caregivers such as spouses or other close relatives. The most important groups of people working in health care settings are physicians, nurses, midwives, pharmacists, administrative staff, technical staff, medical informaticians, or health information management staff and managers. Large facilities, such as university medical centers, are managed by a board of directors.

Within each of these stakeholder groups, different needs and demands on the health information system may exist, depending on the role, tasks, and responsibilities (Sect. 1.3). Ward physicians, for example, require different information than physicians working in service units or in a medical office. Patients sometimes need similar information as physicians but in a different form.

2.8 Functions, Processes, and Entity Types in Health care Settings

In this book, we want to clearly and unambiguously describe the systems needed to ensure information and knowledge logistics. To do this, we need clear concepts to describe the information and knowledge to be provided, the situation in which it is needed, and the people involved. For this reason, we introduce the concepts of *entity*, *entity type*, *function*, *process*, and *role* in this section.

Entities are excerpts of the real or conceivable world.

Think back to the “Russo example” from Sect. 1.4. After his stay in the hospital, Mr. Russo’s GP Dr. Andersson receives discharge letters from both Ploetzberg Hospital and the Kreikebohm Rehabilitation Centre. The “patient Mr. Russo” and the “discharge letter for Mr. Russo from 2020-08-15” are examples of entities.

An *entity type* is the set of virtual or physical entities that have certain properties in common (e.g., “discharge letter” or “patient”).

Entity types form a “unit of thought” when talking about similar entities. Thus, both the discharge letter for Mr. Russo and a discharge letter for another patient, Mrs. Smith, belong to the same entity type “discharge letter” and share the same properties such as sending date, author, and recipient.

For the sake of simplicity, we sometimes take entity types as representatives of the covered entities and their data. If, during certain information-processing activities (e.g., admitting patients), data on entities (e.g., name of patients’ hometown) is used and interpreted, we simply say that the entity type “patient” is used during *administrative admission* of a patient. In this sense, the entity type “discharge letter”

is updated during *medical discharge*. We will also simply say “data on entity type X” if we mean the data describing entities of entity type X (e.g., “data on entity type ‘patient’” means data on patients).

An *information-processing function* is the class of similar activities which update or use entity types. Due to their similarity for all patients in a health care facility, the above-mentioned information-processing activities *administrative admission* and *medical discharge* can be considered information-processing functions.

An *information-processing function* (short: *function*) is a directive in a health care setting on how to use data on entity types and how to update data on entity types. An information-processing function has no definitive beginning or end.

Functions are ongoing and continuous. They describe what is to be done, not how it is done. Functions describe which data on entity types are used to perform the function and which data on entity types are updated by the function.

Functions are performed by human or technical actors.

One can also understand functions as tasks of an actor. But not every task of an actor is an information- processing function. It is only an information-processing function if data on one entity type is used and, after processing this data, the data on another or on the same entity type is updated. For example, a clerk who performs the function *administrative admission* in a hospital needs to ensure that the patient’s administrative data, i.e., the patient’s name, contact data, insurance data, and personal identifiers, are up to date when the patient is admitted to the hospital. The entity type representing patients and their administrative patient data is simply called “patient.” During *patient admission*, the clerk may

- search for (i.e., “use”) the patient’s administrative data among all instances of the entity type “patient” in the *patient administration system*,
- check (i.e., “use”) the patient’s administrative data which is already available if the patient has been treated in the hospital before,
- change (i.e., “update”) parts of the patient’s administrative data if, for example, the address or the insurance data has changed, and
- insert (i.e., “update”) new patient administrative data if the patient is admitted to the hospital for the first time.

Thus, we can state that the function *patient admission* updates and uses the entity type “patient.”

Functions are usually denoted by nouns or gerunds (i.e., words often ending with -ing or -ion), for example, care planning or *patient admission*.

Functions can be structured into a hierarchy of functions, where a function can be described in more detail by refined subfunctions. For example, *nursing admission* can be seen as a subfunction of *patient admission*. There are different opportunities of refining functions that are further described in Sect. 2.14.

An *activity* is an instantiation of a function. For example, “the physician admits the patient Mr. Russo” is an activity of the function *patient admission*. In contrast to functions, activities have a definite beginning and end.

To describe how a function is performed may require not only information about its subfunctions but also information about their chronological and logical sequence. This information is described by *business processes*.

Business processes describe the sequence of activities together with the conditions under which they are performed.

Business processes are usually denoted by verbs which can be followed by a noun (e.g., “admitting a patient,” “planning care,” or “writing a discharge letter”).

Instances of a business process are composed of the individual activities; hence, they also have a definite beginning and end. While functions concentrate on the “what,” business processes focus on the “how” of activities. Functions can be considered representatives of business processes. For example, there is the function *patient admission* and the business process *patient admission* in a hospital. The function *patient admission* is specified by the entity types used and the entity types updated when a patient is admitted. The corresponding business process describes the activities of *patient admission* in their chronological and logical sequence.

For both functions and processes, it is necessary to know who is responsible for them or who performs them. The concept of a role summarizes all the stakeholder groups and groups of people working in health care settings.

Roles describe the sum of expectations addressed to persons or groups of persons.

Roles can also be regarded as a surrogate for the set of functions to be performed by a person or group of persons together with the resulting duties and the rights needed to perform the functions.

Typical roles in health care settings are ward physician, head nurse, project manager, or *chief information officer* (CIO).

The term “information-processing function” presented in this section is related to the term *enterprise function* from business informatics. *Enterprise functions* mainly emphasize the contribution of activities to *business goals*, whereas functions, in the meaning presented here, emphasize the information- processing aspects of activities.

2.9 Application Systems, Services, and Physical Data Processing Systems in Health Information Systems

Whereas functions describe what is done, we now want to look at how information, knowledge, and data processing is done. We will thus take a closer look at tools for data, information, and knowledge processing, in particular *physical data processing systems* and *application components*.

Physical data processing systems ensure the storage, manipulation, and communication of data.

Most people would intuitively call such systems the physically touchable hardware of an information system or physical tool. Physical data processing systems are able to receive, store, forward, or purposefully manipulate data. We denote receiving, storing, forwarding, and purposefully manipulating data as data processing.

Physical data processing systems can be human actors (such as the person delivering the mail), non-computer-based *physical tools* (such as forms for nursing documentation, paper-based patient records, filing cabinets, or telephones), or *computer systems* (such as terminals, servers, personal computers (PCs), or tablets). Computer systems can be physically connected via data wires, leading to physical networks.

A physical data processing system is a physical entity that is able to receive, store, forward, or purposefully manipulate data.

For the administration of physical data processing systems that are computer systems, it can be useful to abstract from single pieces of hardware and instead focus on the optimum use of available processing power, storage, or network capacity. For this reason, the technique of hardware virtualization has found its way into data processing centers in recent years. Virtualization software can help simulate the behavior of servers, storage, and networks. Virtual servers (or virtual machines), for example, simulate the functionalities of physical servers. To install different software products which require different operating systems, virtual servers that run on one physical server can be used. By contrast, in a server cluster, different servers could alternatively, depending on their capacity, run a certain application system. The server cluster, however, can be managed as one (virtual) server.

Virtualization techniques to simulate computer systems are widely used in professional health care settings. When using the term “physical data processing systems” in this book, we include their possible implementation as simulated computer systems, i.e., as simulated physical data processing systems such as virtual machines or server clusters.

A computer system is useless without software. Software can be considered as explicit rules for processing the data in a computer system.

An *application software product* is an acquired or self-developed piece of software that can be installed on a computer system.

By installing and customizing an application software product on a computer system and customizing it to the users' needs, application systems (computer-based application components) are created.

An *application system* is the installation of a certain application software product on a certain computer system. It supports certain functions of a health care setting or communication between other application systems and can store and communicate data on certain entity types.

Application systems may be described by the functions they support and by the *features* they provide. *Features* are functionalities offered by the application software product of the application system which directly contribute to the fulfillment of one or more functions. The finer the granularity of a function, the greater is the probability that the function semantically corresponds to a feature offered by an application system.

We denote features by a short phrase consisting of at least one verb and one noun expressing the ability of the application software product.

For example, the application system *patient administration system* stands for the installed application software product to support the functions *patient admission* and *administrative discharge and billing* in a hospital. It may offer the features “generate a unique *patient identification number*” (PIN) and “provide catalog of diagnoses” in order to fulfill the functions *patient admission* and *administrative discharge and billing*. Other typical application systems are the *medical documentation and management system (MDMS)*, the *computerized provider order entry (CPOE) system*, and the *picture archiving and communication system (PACS)*. These and other application systems are discussed in more detail in Sect. 3.4. Application systems store data in database systems. Depending on the architecture style of a health information system, each of its application systems either has its own database system or uses the database system of another application system (Sect. 3.5).

Even in highly computerized health care settings, not every information-processing function is supported by an application system. Sometimes, the rules for processing data are not implemented as executable application software product but as organizational rules or working plans that describe how people use certain physical data processing systems. For example, the rules regarding how, by whom, and in which context given forms for nursing documentation have to be used in a certain hospital may be described verbally as text in a handbook of this hospital. In this example, the paper-based forms that are used represent physical data processing systems. We call sets of organizational rules for data processing which are

implemented by non-computer-based physical tools “non-computer-based application components.” They are often also denoted as *paper-based application components*.

“Application component” is an abstract concept for both application systems and non-computer-based application components.

An *application component* is a set of implemented rules which control data processing of certain physical data processing systems. It supports certain functions of a health care setting or communication between application components.

For those dealing with the management of an information system, it is important to have an overview of the information system’s application systems. However, users often do not know which application systems they are using. They are merely interested in certain features provided on a website or by an app on their smart-phone. The actual application system providing these features may even be hidden from the users and invoked by another application system. We call these features that are provided by one application system for use by another application system and which are thus not immediately used by users “services.” Using a service means invoking it.

A *service* is an encapsulated feature provided by application systems in order to be invoked by other application systems.

Details on the most relevant tools for data, information, and knowledge processing, i.e., application components, services, and physical data processing systems, in health care settings can be found in Sect. 3.4 and the following sections.

2.10 Electronic Health Records as a Part of Health Information Systems

The most important functions of health care settings are related to *prevention*, diagnostics, therapy, and rehabilitation. Obviously, data and documents that are relevant to medical decision-making both in diagnostics and in therapy need to be collected and presented in a record for the patient.

Until just a few years ago (and in some cases still in the present), many documents in the records have been paper-based documents, such as laboratory results or discharge summaries. The portion of documents created and stored in computer-based application systems has increased, however, in recent years and will continue

to increase further. It therefore seems natural to strive for a record that is used and updated by application systems and stored in database systems: *the electronic health record (EHR)*.

The *electronic health record (EHR)* is the collection of a person's health data from different health care settings. It is stored by one or more application systems in a transinstitutional health information system (tHIS).

This means that the EHR for a person might be scattered physically across the database systems of multiple (discrete or interconnected) application systems at various health care facilities. Each of the database systems will hold and manage a partial EHR containing partial patient information or, to be more precise, containing data about patient-specific entity types. Each partial EHR is scoped according to the person's stays at the health care settings which will be discussed in Sect. 3.5.

EHRs provide relevant information about a person whenever and wherever it is needed during *patient care*. Furthermore, EHRs provide information that is relevant for administrative functions, such as *billing* and *quality management*.

An *electronic patient record (EPR)* is the collection of a person's health data from one certain health care facility where the person is or has been a patient. They are stored by application systems designated for this purpose by the facility.

Some years ago, EPRs were the predominant form of electronic records in health care. Hence, potentially relevant information about the medical history of a patient that was recorded in one facility was missing or had to be recorded again in another facility. This led to quality and efficiency problems.

Although this situation can still be found in many facilities, efforts are being made today to organize EPRs as patient-centric, i.e., independent of boundaries of facilities, which will transform the multiple EPRs to one EHR for one person.

Different strategies can be found to achieve the vision of a complete and lifetime-spanning EHR that supports health care on the one hand but respects legal and ethical issues on the other. These are described further in Chap. 3.

In the international literature, the terms EHR and EPR are usually defined as presented here. In some countries, however, the use of these terms may differ. According to the German data privacy law, for example, health insurers are obliged to provide their insured persons with the so-called electronic patient record (EPR) which contains selected patient data from different facilities. This "EPR," in fact, corresponds more to our definition of an "EHR."

2.11 Architecture and Infrastructure of Health Information Systems

The *architecture* of an information system describes its fundamental organization, represented by its components, their relationships to each other and to the environment, and by the principles guiding its design and evolution [2].

The architecture of an information system can be described by functions, business processes, application components, services and physical data processing systems, and their mutual relationships.

There may be several architectural views of an information system, e.g., a functional view looking primarily at the functions or a process view looking primarily at the business processes. Architectures that are equivalent with regard to certain characteristics can be summarized in a certain *architectural style*.

The set of components of the information system and services, which are centrally coordinated and provided for use throughout the health care setting, is called the *infrastructure of an information system*. The infrastructure of an information system consists of physical data processing systems such as servers set up centrally in a data center as well as printers and scanners made available for all users at central locations. The infrastructure may also contain *logical tools* such as central application systems that have to be used by most of the users throughout the health care setting. Moreover, the service desk providing support for all users in the health care setting is also part of the infrastructure. Components and services that are dedicated only to a specific department are not considered to be part of the infrastructure [3].

2.12 Management of Information Systems

Information systems need systematic management. In general, management comprises all leadership activities that determine the goals, structures, and behaviors of a setting. Management as a task includes *planning*, *directing*, and *monitoring* a specific object. Within a setting, management can focus on different aspects and objects of the setting. In companies, for example, a distinction is made between management of finances and management of personnel. Accordingly, there is the management of a setting's information system.

Management of information systems (short: information management) means

- planning the information system and its architecture,
- directing its construction and the further development of its architecture and its operation on the basis of these plans, and
- monitoring compliance of its development and operation with the plan specifications.

The goal of managing information systems is systematic information processing that supports information and knowledge logistics and therefore contributes to the setting's strategic goals (such as efficient *patient care* and high satisfaction of patients and staff in a health care setting). *Management of information systems* therefore directly contributes to the setting's success and the ability to compete.

Management of information systems encompasses the management of all components of the information system—the management of functions, processes, and entity types, of application components and services, and of physical data processing systems.

Management of information systems is discussed in detail in Chap. 4.

2.13 Modeling Information Systems

Modeling health information systems is an important precondition for their management: What we cannot describe, we usually cannot manage adequately. But what is a model?

A *model* is a description of what the modeler believes to be relevant about a system.

In the sciences, models commonly represent simplified depictions of reality or excerpts of it. Models are adapted to answer certain questions or to solve certain tasks. Models should be appropriate for the respective questions or tasks. This means that a model is only “good” when it is able to answer such a question or solve such a task. For example, a model that only comprises the patients (and not the nurses) of a ward cannot be used for nurse staffing and shift planning. Since we are dealing with *management of information systems*, this means that models should present a simplified but appropriate view of a health information system in order to support *management of information systems*.

Examples of respective questions that can be answered by specific information system models could be:

- Which functions are supported by computer-based logical and physical tools?
- Which tools for data, information, and knowledge processing are used in a nursing home?
- What information is needed if a patient is to be admitted to a rehabilitation hospital? What information can be provided afterwards?
- Which functions of a hospital are affected in the event that a specific server breaks down?
- How can the quality of information processing in a regional health care network be judged?

The better a model assists its users in answering a given question, the better the model is. Thus, the selection of the adequate model depends on the problems or questions to be answered or solved.

There exists a large number of different classes of models. Each class of models is determined by a certain metamodel. A metamodel can be understood as a language for constructing models of a certain class and a guideline for using the language.

A metamodel is a modeling framework which consists of

- modeling syntax and semantics (the available modeling concepts together with their meaning), i.e., the modeling language;
- the representation of the concepts (how the concepts are represented in a concrete model, for example, in a graphical way); and
- (sometimes) the modeling rules (e.g., the modeling steps), i.e., the guideline for applying the language.

Just as there are different views on health information systems, there also exist various metamodels. Typical types of metamodels are as follows:

1. Functional metamodels focus on functions (Sect. 2.8) that are supported in a health information system. They provide the means to describe dependencies between functions, for example, hierarchies of functions or information flows between them.
2. Technical metamodels are used to build models describing the tools for data, information, and knowledge processing (see, for example, application components, physical data processing systems, Sect. 2.9) used in a health care setting. They also help to describe data transmission or *communication links* between the tools. If the model comprises a graphical presentation of tools and their links, it also visualizes the architecture of a health information system. Examples of technical metamodels are technical network diagrams or application landscape diagrams.
3. Organizational metamodels help to describe the organizational structure of a health care setting. Organizational models typically consist of *organizational*

units or roles and their hierarchies. Examples of organizational metamodels are organizational charts (also called organigrams).

4. Data and information metamodels are used for building models of the structure of data and information processed and stored inside health information systems. Their concepts are typically entity types and their relationships. Examples of data metamodels are UML class diagrams (UML = Unified Modeling Language) or entity-relationship models (ERMs).
5. Business process metamodels focus on a dynamic view of information processing in health care settings. They provide concepts that describe the activities to be done, their chronological and logical order, the conditions under which they are performed, and often their links to roles, organizational units, entity types, and logical or physical tools for data and information processing. Examples of business process metamodels comprise UML activity diagrams, event-driven process chains (EPCs), Petri nets, or the business process modeling and notation (BPMN) language.
6. Information system metamodels (also: enterprise metamodels) combine different metamodels (i.e., functional, technical, organizational, data, or business process models) into an integrated, enterprise-wide view on information processing in a facility. Examples of information system metamodels comprise the *three-layer graph-based metamodel (3LGM²)* (Sect. 2.13), The Open Group Architecture Framework (TOGAF), the Extended Enterprise Modeling Language (EEML), or the Architecture of Integrated Information Systems (ARIS).

Modeling of health information systems is based on the right selection of a metamodel. For health information system modeling, you should therefore consider the following steps:

1. Define the questions or tasks to be solved by the health information system model.
2. Select an adequate metamodel.
3. Gather the information needed for modeling.
4. Create and validate the model.
5. Analyze and interpret the model (answer your questions).
6. Evaluate if the right metamodel was chosen, i.e., if the model was adequate to answer the questions. If not, return to step 2.

Especially step 3 of gathering the information needed for modeling is often time- and cost-intensive.

Modeling patterns which can be customized to the specific situation to be modeled can reduce the modeling effort. We call these types of models *reference models*. According to the metamodel used, a reference model supports the construction of models of a certain class of systems and helps to deal with a certain class of questions or tasks concerning these systems.

A model is called a *reference model* for a certain class of systems and a certain class of questions or tasks dealing with these systems if it provides model patterns supporting

- the derivation of more specific models through modifications, limitations, or completions (generic reference models) or
- direct comparison of different models with the reference model concerning certain quality aspects of the modeled systems (e.g., completeness, styles of system's architecture) (non-generic reference models).

A specific model may be considered a variant of a generic reference model developed through specialization (modifications, limitations, or completions). This variant is an instance of the metamodel that also underlies the corresponding reference model. For example, a model of the processes in a hospital information system of a specific hospital may be derived from a general reference model on health information system processes. Both the specific model and the reference model used are instances of the same business process metamodel.

A reference model should be followed by a description of its usage, for example, how specific models can be derived from the reference model or how it can be used for the purpose of comparison.

Specific models can be compared with a reference model, and consequently models can also be compared with each other, judging their similarity or difference when describing certain aspects.

Reference models can be normative in the sense that they are broadly accepted and have practical relevance. Reference models are more likely to be accepted if they are not only reliable and well-tested but also recommended by a respected institution. For example, the initiative Integrating the Health care Enterprise (IHE) (Sect. 3.7.2.5) provides a comprehensive set of models describing how to use communication standards such as Health Level 7 (HL7) and Digital Imaging and Communications in Medicine (DICOM) in typical health care settings. These models can be regarded as reference models. Many experts in the field use these reference models as norms or standards although they are explicitly not. These models apparently became normative because they are widely used especially in commercial invitations of tenders for software supporting radiology departments.

In the following section, we introduce the 3LGM² as an information system metamodel that integrates aspects of functional metamodels, technical metamodels, organizational metamodels, and data metamodels. For 3LGM², there are also reference models describing certain aspects of health information systems available (Sect. 3.11.1).

2.14 3LGM²: A Metamodel for Information System Architectures

The *three-layer graph-based metamodel (3LGM²)* is a metamodel for modeling (health) information systems. It aims to support the systematic management of health information systems and especially the structural quality assessment of information processing in health care settings. We will use this metamodel further on in this book (especially in Chaps. 3 and 6) and thus present it in detail here.

Typical questions to be answered with models derived from the 3LGM² metamodel are as follows:

- Which functions of a health care setting are supported?
- Which information is needed or updated when performing a function?
- Which application components are used and how do they communicate?
- Which physical data processing systems are used?
- Which functions are supported by which application component?
- Which application components are installed on which physical data processing systems?
- What is the overall architecture of the health information system?

3LGM² combines functional, technical, and organizational aspects with certain aspects of data and process metamodels. As the name indicates, the 3LGM² distinguishes three layers of information systems:

- domain layer,
- logical tool layer,
- physical tool layer.

The following sections provide a user-oriented description of the three layers, complemented by examples from health information systems.

2.14.1 Domain Layer

The *domain layer* of a 3LGM² model describes what kinds of activities in a health care setting are enabled by its information system and what kind of data is stored and processed. This layer is independent of the implemented physical and logical tools for data and information processing.

Information-processing activities at a certain time and place in an information system use certain data in order to create, update, or delete other data. For example, the clerk entering Mr. Russo's administrative data into the *patient administration system* when he arrives at the Kreikebohm Rehabilitation Centre creates or updates Mr. Russo's patient data. For the sake of simplicity, we will from here on subsume creating, updating, or deleting patient data under the term "updating."

In Sect. 2.8, we already introduced the important concepts for the domain layer, namely entities, entity types, and information-processing functions. Entities are excerpts of the real or conceivable world, such as “patient Mr. Russo,” while an entity type (such as “patient”) is a set of virtual or physical entities that have certain properties in common. An information-processing function (short: function) is a directive in a health care setting on how to use data on entity types and how to update data on entity types (such as *care planning* or *patient admission*). At the domain layer, we now use these concepts for health information system modeling to describe entity types, functions, and the relationships between functions and entity types performed in a health care setting.

Figure 2.2 shows an example. The function *administrative admission* updates the entity type “patient,” which represents a patient’s administrative data. This indicates that during the *administrative admission*, patient data such as name, birthdate, insurance data, and identification numbers are documented for the first time or updated. The entity type “patient” is used by the function *medical admission* which indicates that during a *medical admission*, the administrative data is available and can be used. *Medical admission*, in turn, updates the patient’s “medical history.” This indicates that information on the medical history is documented or updated during *medical admission*. Both the entity types “patient” and “medical history” are needed to create a medical care plan. Therefore, these two entity types are used by the function *medical care planning*. In 3LGM² models, functions are represented by rectangles and entity types are represented by ovals.

Functions and entity types can be structured hierarchically by “specialization” and “decomposition.” When a function or an entity type is specialized, all its sub-elements are a refinement of the function or the entity type and independent of the respective super-element. For a function, this means that the activities regarding this function are performed differently in different contexts. The function “execution of

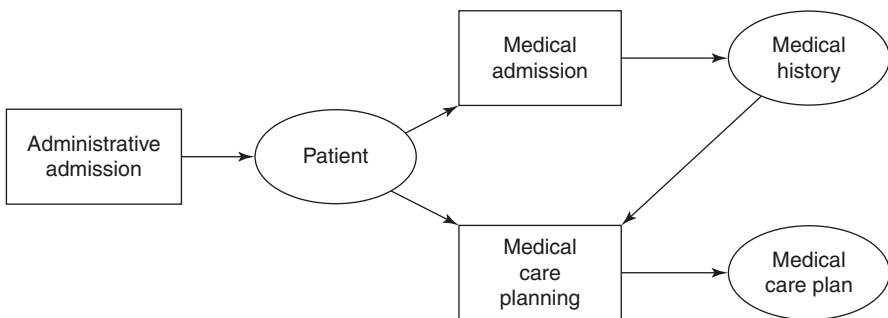
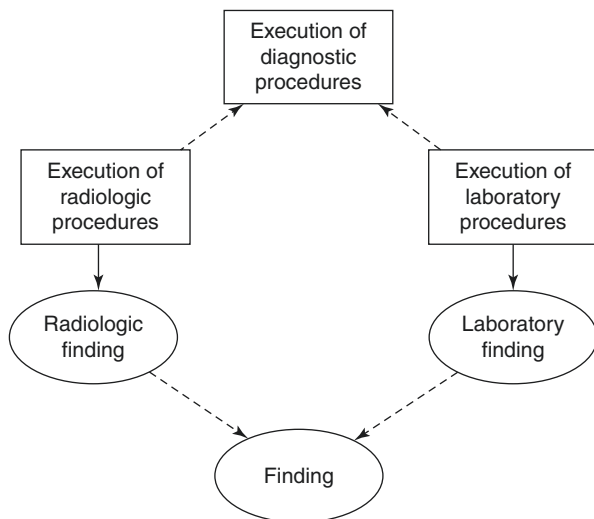


Fig. 2.2 3LGM²² representation of functions (represented by rectangles) and entity types (represented by ovals) at the domain layer of 3LGM²². An arrow pointing from a function to an entity type represents an updating access of an entity type. An arrow pointing from an entity type to a function represents a using access of an entity type

Fig. 2.3 3LGM² representation of a specialization of functions and entity types. “Execution of radiologic procedures” and “execution of laboratory procedures” are specializations of the function “execution of diagnostic procedures.” “Radiologic finding” and “laboratory finding” are specializations of “finding”



diagnostic procedures,” for example, has different specializations in different diagnostic departments. Similarly, an entity type can have different forms for slightly different purposes: A radiologic finding is different from a laboratory finding; but both are specializations of findings, which is the generalized term (Fig. 2.3).

By contrast, when a function or an entity type is decomposed, all its sub-elements form a proper subset of the function or the entity type. An activity regarding a function is only completed if all activities regarding all its decomposed subfunctions are completed. For example, the activities regarding *patient admission* are only completed if *appointment scheduling*, *patient identification*, *administrative admission*, *medical admission*, *nursing admission*, and *visitor and information services* have been performed (Fig. 2.4). Similarly, a decomposed entity type is only complete when all its subordinate entity types are available. The entity type “patient,” for example, must contain a name, a PIN, the patient’s address, and insurance data.

Both decomposition and specialization are represented by dashed arrows from sub-elements to super-elements in 3LGM². For modelers, it is important to differentiate between specialization and composition at the domain layer. To avoid misunderstandings, it might be useful to predefine the use of only one hierarchical relationship for functions or entity types in one model. If this is not possible, one should at least consider that an entity type or a function cannot be specialized and decomposed at the same time.

Using relationships and updating relationships between functions and entity types are inherited to their sub-elements, no matter whether the functions or entity types were decomposed or specialized. This means, for example, that the PIN, which is a sub-element of the entity type “patient,” may be updated by the functions *patient identification*, *administrative admission*, etc. although the “update” relationship is only modeled between the super-ordinated entity type “patient” and the respective functions.

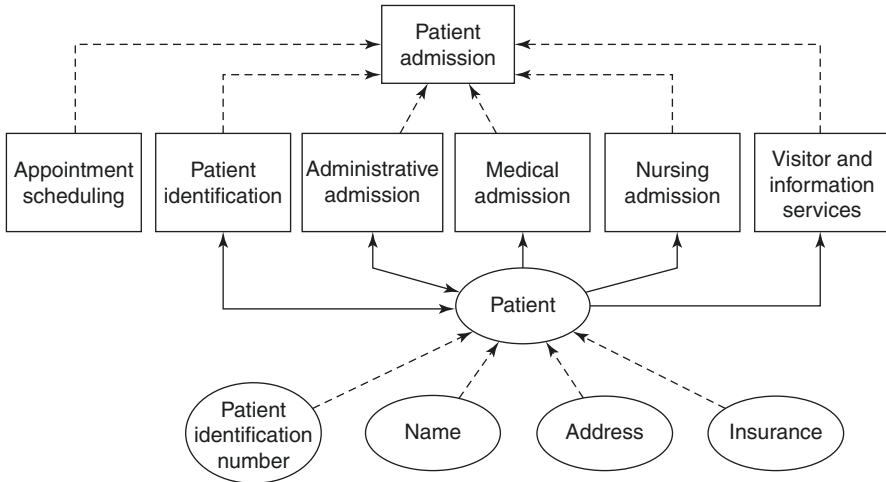


Fig. 2.4 3LGM² representation of a decomposition of the function *patient admission* and of the entity type “patient”

Functions are usually performed in certain parts of health care settings. The execution of radiologic procedures, for example, is performed in the radiology department of a hospital. We call those parts of health care settings “organizational units.”

An *organizational unit* is a part of a facility which can be defined by responsibilities.

Organizational units such as a radiology department can be decomposed, but not specialized.

Functions, entity types and organizational units are part of a static view of a health care setting. For modeling the dynamic view of health care settings, business process models are more appropriate. Which entity types and which functions of an information system are modeled depends on the health care setting and on the modeling purpose. Reference models may offer recommendations on important entity types and functions for certain kinds of hospitals.

2.14.2 Logical Tool Layer

2.14.2.1 Application Systems

At the *logical tool layer*, application systems or, in a broader sense, application components are the center of interest. As defined in Sect. 2.9, an application system is the installation of a certain application software product on a certain computer

system. Application components as a more general concept are sets of implemented rules that control data processing of certain physical data processing systems. Application systems as well as non-computer-based application components support functions.

An application system cannot be bought in a shop but must be constructed by customizing a buyable application software product onsite. A *patient administration system*, for example, is implemented by an application software product offering features for *appointment scheduling*, *patient identification*, checking for readmitted patients, and *administrative admission*. Many application software products developed for health care facilities consist of different modules, and buyers can decide which of the modules of the application software product they purchase. Application software products for *enterprise resource planning systems (ERPS)*, for example, may offer modules for *human resource management*, *material management*, *financial accounting*, and *customer or patient administration* (Fig. 2.5).

Non-computer-based application components are controlled by rules which we can understand as working plans describing how people use non-computer-based data processing systems to support a given function. A working plan may be available in written form in a document (e.g., in an SOP—standardized operating procedure). In most cases, however, working plans are verbal agreements or are merely specified implicitly. A non-computer-based application component for patient data privacy forms, for example, may be controlled by a working plan that describes when and how to hand out paper-based data privacy forms to the patients and where and how to archive the signed document.

Application components of an information system are objects that are recognized and used by staff members in a facility. But nevertheless, they are not tangible in the same way as physical tools are. We therefore refer to application components also as *logical tools*. Consequently, we call the layer describing the application components the logical tool layer. This is in contrast to the tangible tools, which we refer to as physical.

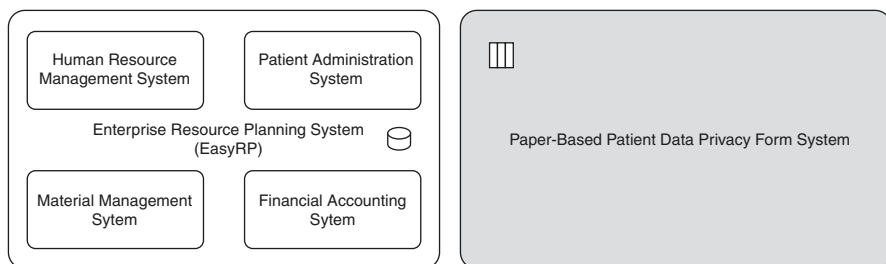


Fig. 2.5 3LGM² representation of the application system *enterprise resource planning system* which consists of four sub-application systems and a database system and of the paper-based “patient data privacy form system” as non-computer-based application component

At the logical tool layer, application components are responsible for supporting functions and for storing and communicating data on certain entity types. *Computer-based application components* usually store data in database systems which are controlled by database management systems. Non-computer-based application components use document collections for data storage.

Both application systems as well as non-computer-based application components are represented by rounded rectangles at the logical tool layer of a 3LGM² model. Visually they can be distinguished by different coloring and the different symbols for database systems (cylinders) and data collections (dashed rectangles, Fig. 2.5).

For communication between two application systems, we distinguish between the message-oriented and the service-oriented communication paradigm, which are explained in the following sections.

2.14.2.2 Message-Oriented Communication

For message-oriented communication, application systems use *communication interfaces*. A communication interface can either send or receive messages over communication links. A *patient administration system*, for example, may communicate with an *MDMS* by sending messages over communication interfaces and a communication link (Fig. 2.6). In this example, the message may comprise information on the admission of a patient and the related administrative patient data.

A *message* is a set of data on entities (e.g., administrative data on a given patient) that are arranged as a unit in order to be communicated between application systems. A message type describes a class of uniform messages and determines which data on which entity types is communicated by a message belonging to this message type. For example, the message type “patient administrative data” could describe how the administrative data on a patient (name, address, identification number, insurance data, etc.) must be arranged in a uniform way in order to be understood by both the *patient administration system* and the *MDMS*.

A message type can belong to a communication standard, i.e., a standard for *syntactic interoperability* (Sect. 3.7.1). There are several communication standards which describe how messages of a certain data format must be communicated between application systems. In medical informatics, Health Level 7 Version 2 (HL7 V2) and DICOM are well-known examples of such message-oriented



Fig. 2.6 3LGM² representation of a *patient administration system*’s communication interface (represented by a circle) sending messages to a *medical documentation and management system*’s communication interface over a communication link (represented by an arrow)

communication standards (Sects. 3.7.2.1 and 3.7.2.4). Application systems have to communicate by using their interfaces to ensure that functions can use and update entity types as described at the domain layer.

The concept of a message-oriented communication paradigm may also be used to model the communication between application systems and non-computer-based application components.

2.14.2.3 Service-Oriented Communication

The service-oriented communication paradigm assumes that application systems provide encapsulated features (“services”) that can be used by other application systems. A *patient administration system*, for example, could offer a service “get patient” to other application systems within a health care facility. When invoking this service, an application system such as the *MDMS* can request and obtain the administrative patient data of a given patient from the *patient administration system*.

Application systems need “providing interfaces” to provide services to other application systems and “invoking interfaces” to invoke services provided by other application systems. In 3LGM² models, invoking interfaces are represented by circles and providing interfaces are represented by triangles (Fig. 2.7).

Services themselves are not graphically represented at the logical tool layer but can be assigned to interfaces. Services of a similar type can be summarized in 3LGM²² service classes.

A function can be either supported by one service or by a set of combined (“orchestrated”) services. In health information systems, the *interoperability standards* HL7 FHIR (Fast Health care Interoperability Resources) and open Electronic Health Record (openEHR) support the implementation of *service-oriented architectures* (SOA).

Communication with non-computer-based application components can take different forms and is therefore considered separately.

Communication of data between two non-computer-based application components is only possible through an active human intervention, for example, by carrying a paper document from one place to another.

In a similar way, human intervention is necessary for the communication between non-computer-based application components and application systems. Scanning a paper form for archiving or typing a discharge letter which is available as an audio recording are examples of communication from a non-computer-based application



Fig. 2.7 3LGM²² representation of a situation where the *patient administration system* provides a service “get patient” invoked by the *medical documentation and management system*

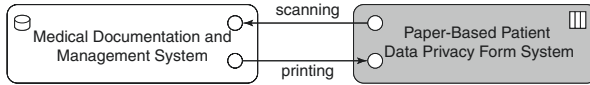


Fig. 2.8 Communication between an application system and a non-computer-based application component

component to an application system. Printing out a paper form available in the *MDMS* or storing radiological images on a memory device to be taken to another health facility by a patient are examples of communication from an application system to a non-computer-based application component. Figure 2.8 shows the 3LGM² representation of such “media breaks” between application systems and non-computer-based application components. The details of the communication should be modeled by messages (Sect. 2.14.2.2).

2.14.3 Physical Tool Layer

The *physical tool layer* describes physical data processing systems and their data transmission links among each other. As defined in Sect. 2.9, a physical data processing system is a physical entity that is able to receive, store, forward, or purposefully manipulate data.

For the computer-based part of information systems, servers, PCs, notebooks, tablets, switches, routers, smartphones, etc. are modeled at the physical tool layer. In addition, virtualized physical data processing systems are modeled at the physical tool layer because they behave like physical data processing systems to the outside world (Sect. 2.9).

For the non-computer-based part of information systems, human actors (such as persons delivering mail) and non-computer-based physical tools (such as printed forms, telephones, books, paper-based patient records, administrative stickers) are modeled at the physical tool layer.

Figure 2.9 shows a simple model of the physical tool layer. A virtualized server farm is represented by one physical data processing system. This “black box” is connected to a patient terminal, a PC and a tablet PC. The physician uses both a tablet PC and a telephone as physical computer-based or non-computer-based tools, respectively.

Depending on the modeling goals, health professionals, patients, or caregiving relatives can be modeled as physical data processing systems (as in Fig. 2.9) to highlight their information-processing role in the health information system. In most cases, this will not be necessary.

To specify the relationship between physical data processing systems and virtualized physical data processing systems in a 3LGM² model, a “virtualizes” relationship can be modeled. Figure 2.10 illustrates the 3LGM² representation of

Fig. 2.9 3LGM² representation of physical data processing systems at the physical tool layer. The patient terminal, the PC and the tablet PC are connected with the virtualized server farm

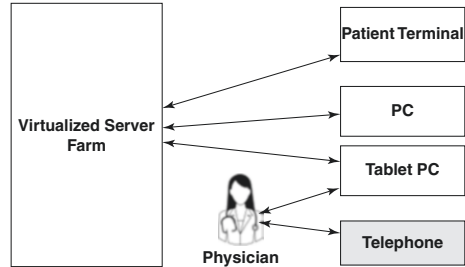
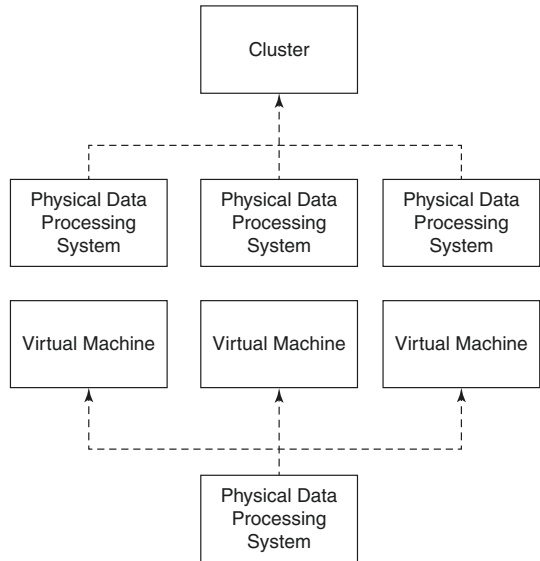


Fig. 2.10 3LGM² representation of a virtual machine. On the top, the concept of a cluster virtualizing several physical data processing systems is illustrated. At the bottom, there is one physical data processing system which is virtualized into several virtual machines



virtualization techniques. In a server cluster, physical data processing systems can run certain application systems alternatively. Virtual machines allow multiple operating systems or different instances of one operating system to run on one physical data processing system (compare Sect. 2.9).

Physical data processing systems such as a specific server or a specific PC can be assigned to a “tool class” (e.g., server, PC) and a location.

Physical data processing systems are physically connected via data transmission links (e.g., communication network, courier service) which can use different transmitting media. A transmitting medium is either signal-based (e.g., copper cable, optical fiber) or non-signal-based (e.g., sheet of paper, CD-ROM, USB flash drive).

Physical data processing systems can be refined by decomposition. A physical data processing system can be part of exactly one physical data processing

system. Thus, the lower part in Fig. 2.10 does not show a decomposition but a virtualization.

2.14.4 *Inter-layer Relationships*

A variety of dependencies, called inter-layer relationships, exist among concepts of the three layers of a 3LGM² model. Relationships exist between concepts of the domain layer and the logical tool layer and between concepts of the logical tool layer and the physical tool layer.

The following relationships between the domain layer and the logical tool layer can be modeled in 3LGM²:

- Functions (domain layer) can be supported by application components or, in SOA, by services which are both modeled at the logical tool layer.
- Entity types (domain layer) or, more exactly, their representation by a dataset or document collection, can be stored in an application component (logical tool layer).
- An application system storing an entity type can, in addition, be the primary application system of that entity type. This means that the application system contains the “original” data on that entity type. Data on that entity type that are stored in other application systems have to be considered copies of the “original” data. Consequently, only data in primary application systems can be updated directly by users; *data integrity* in the other application systems must be maintained by sending new copies of the original data to the other application systems (see Sects. 3.5 and 3.8.1 for an in-depth discussion of data integrity and *data integration*).
- In the message-oriented communication paradigm, entity types or, more exactly, their representation by a message can be communicated over communication interfaces and communication links.
- In the service-oriented communication paradigm, entity types are represented by parameters that are handled by services.

Between the logical tool layer and the physical tool layer, there are two types of relationships expressing that application components need certain physical data processing systems to work:

- One relationship between application components and physical data processing systems states that an application component needs physical data processing systems to be able to provide its features. For example, an application system needs to be installed on a server to make its features available.
- The second relationship between application components and physical data processing systems states that application components need a physical data processing system to store data on entity types.

2.14.5 First Steps of 3LGM² Modeling

2.14.5.1 Installation of the 3LGM² Tool

To start modeling, the current version of the full version of the 3LGM² tool can be downloaded from <http://www.3lgm2.de>. The Java-based tool runs on different platforms and can freely be used for non-commercial purposes.

2.14.5.2 Modeling the Domain Layer

The main elements of the domain layer are entity types and functions. When modeling a domain layer from scratch, the following rules should be observed:

- Each function should use and update at least one entity type.
- Very similar or even equivalent functions that are performed in different areas, different organizational units, or different health care facilities of a health care network should be modeled only once at the domain layer. The functions can be identified as similar by checking whether they use and update the same set of entity types.
- If an entity type is updated or used by a function that is decomposed or specialized, then all of the subfunctions also use or update the entity type. For clearer models, it can be helpful to assign entity types only to functions that are not further refined by subfunctions.
- Functions should be decomposed or specialized only to that level of detail needed to describe the support of the functions by single application components.
- “Documentation” may not be modeled as a function in 3LGM² because it is an inherent part of a function updating an entity type. If an entity type is updated by a function and this entity type’s data are stored in an application component, we call this combination the “documentation of the entity type.” However, sometimes it may improve the readability of a model to include the word “documentation” in a function’s name.

Identifying appropriate functions and entity types for a specific health care setting is a non-trivial task. The most elaborate but also the most direct way to identify functions and entity types of a setting is to conduct interviews with the persons performing the functions. Preparing and conducting these interviews, function patterns, or reference models providing lists of typical functions and of relationships between the functions of a specific type of health care setting may be helpful. In Chap. 3, we develop patterns for functions that are performed in many health care settings. These patterns as well as a reference model for the domain layer of hospital information systems are available at <http://www.3lgm2.de> and can be used and refined for modeling a specific health information system.

2.14.5.3 Modeling the Logical Tool Layer

The logical tool layer describes the application components of a health care setting and the communication between these components. For modeling the logical tool layer, the following rules should be observed:

- Application systems are installations of application software products. For every installed instance of an application software product, there should be one application system in a 3LGM² model.
- Application software products in health care often consist of different modules for different functional areas (e.g., a module for operation planning and execution, a module for nursing). Thus, an application system may consist of sub-application systems (modeled by part-of relationships) according to the installed modules of an application software product.
- For message-based communication between two application systems, each application system should have at least one communication interface. The message type communicated over a communication link between two communication interfaces should be named according to the message type of the communication standard used. If the technical name of the message type is not known or the message type is proprietary, the name of the message type should describe the message type's content concisely, for example, by using the name of the entity type connected to the message.
- Depending on the modeling scope, the modeler may decide to model only application systems or to model a mix of application systems and non-computer-based application components.
- There are at least two strategies for modeling non-computer-based application components. (1) All non-computer-based information processing of the logical tool layer of a health care setting can be modeled with the help of one non-computer-based application component having several communication links to application systems of the setting. (2) For each application system where there is an intervention of a person necessary to enter data which are already documented at another medium, a non-computer-based application component describing this media break is modeled. Likewise, for each application system where intervention by a person is necessary to transfer data stored in the application system to another medium, a non-computer-based application component describing this media break is modeled.

To obtain a correct representation of a logical tool layer, it is in most cases not sufficient to interview health care professionals who work within the health information system. They often have too little insight into the technical details of the logical tool layer. Interviewing information management staff or analyzing the current documentation of the information system are the most promising methods for obtaining information about the logical tool layer of a health care setting.

2.14.5.4 Modeling the Physical Tool Layer

Physical data processing systems and their connections are modeled at the physical tool layer. This layer has the fewest modeling rules. Depending on the purpose, the modeler must decide whether to model single physical data processing systems such as servers and PCs or to provide a more abstract view, for example, by modeling the data processing center of one facility as one physical data processing system.

Information about the physical data processing systems and the network can be obtained from the staff of the information management department or the data processing center, respectively.

2.14.5.5 Modeling Inter-layer Relationships

Functions are to be connected with application components supporting them in a health information system. To establish this relationship between the domain layer and the logical tool layer, the organizational unit where the function is supported by the application component should be specified. This is especially important if a function is supported by one or more application components in a health care setting. Therefore, even if one of these application components fails, this function could still be performed, at least in selected organizational units. In this case, we have a functional redundancy which may be an indication for superfluous application components.

There are also desired functional redundancies. To update the application software product of an application system, it might be necessary to shut down an application system for a few hours. If this concerns an application system which is permanently in use, such as an *MDMS*, it can be helpful to have an evasive application system.

However, we must also be careful that supposed functional redundancy does not result from inaccurate modeling.

In a 3LGM² model, we specialize or decompose functions to that level of detail needed to describe the support of the functions by single application components. That means if we think of the hierarchy of functions in a 3LGM² model as a tree in graph theory, then each of the tree's "leaf functions" must completely be supported by one application component of the information system. We only assign application components to the "leaf functions" of the tree. For example, if we find that the function *medical and nursing care planning* needs joint support of two application components X and Y, we have to specialize or decompose the function in such a way that the resulting subfunctions are supported by X and Y, respectively. If X is used by clinicians and Y is used by nurses, a solution could be to decompose the function into *medical care planning* and *nursing care planning*.

Besides the relationship between functions and application systems, it is important not to forget to model the relationships between entity types of the domain layer and their representation forms at the logical tool layer (message types, parameters, dataset types).

Between the application systems at the logical tool layer and the physical data processing systems at the physical tool layer, the relationships have to be modeled—both for expressing the installation of an application component on a physical data processing systems and for the storage of data in such a system.

2.15 Example

For this example, we merge many of the small examples of Sect. 2.14 into one 3LGM² model showing a section of the information system of a fictional hospital. Figure 2.11 illustrates which logical and physical tools are used for the function *patient admission* in the hospital. Four subfunctions of *patient admission* (*appointment scheduling*, *patient identification* and checking for readmitted patients, *administrative admission*, and *visitor and information services*) are supported by the *patient administration system*, which is a part of the *ERPS*. *Medical admission* and *nursing admission* are supported by the *MDMS*. Obtaining consent for processing of patient-related data is supported by the non-computer-based application component for patient data privacy forms. This application component is based on paper forms which are scanned by a clerk (see physical tool layer) and then stored in the *MDMS*.

The *patient administration system*, which is the master application system (Sect. 3.9.1) for the entity type “patient,” sends the administrative patient data as a message to the *MDMS*. The *MDMS* can thus store this information about the entity type “patient” in its own database; administrative patient data that is needed to support *medical admission* and *nursing admission* as functions therefore do not have to be reentered in the *MDMS*. The entity type “patient” is both stored in the database systems of the *ERPS* and the *MDMS* what is represented by dashed lines between the domain layer and the logical tool layer in Fig. 2.11.

Both the *patient administration system* and the *MDMS* are run on servers at a virtualized server farm (see relationships between logical and physical tool layer). The application systems can be accessed by different end devices (patient terminal, PC, tablet PC).

Note that Fig. 2.11 shows a model of the information system expressing what the modeler believed to be relevant about the information system. It therefore simplifies some aspects which might be relevant in other contexts.

Another visualization of relationships between 3LGM² model elements is the matrix view. Figure 2.12 shows connected functions (columns) and application components (lines) expressing that the functions are supported by certain application components. The *patient administration system* supports three different functions, the *MDMS* supports two functions, and one function is supported by the paper-based patient data privacy form system. The matrix view also helps to identify incomplete parts of models. In Fig. 2.12, we can see that there are no functions modeled that are supported by the *financial accounting system*, the *human resources management system*, and the *material management system*, which are parts of the *ERPS*.

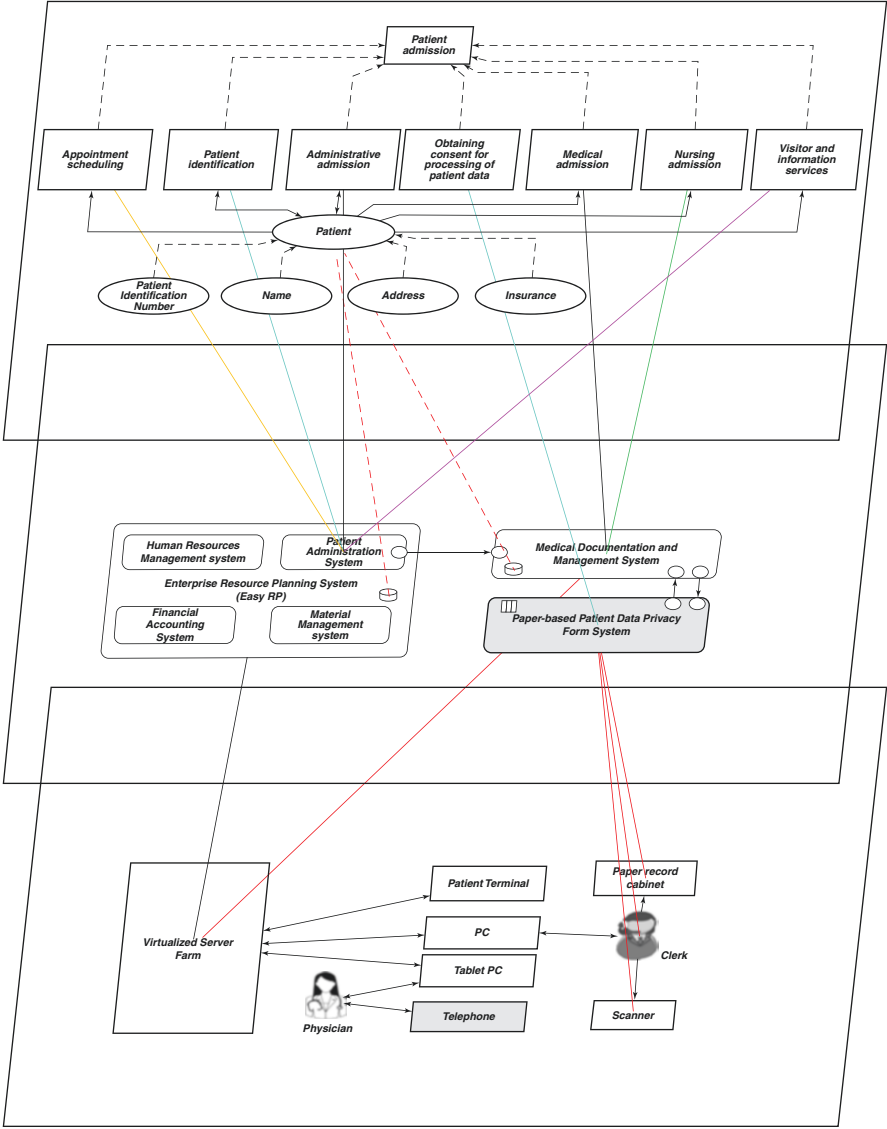


Fig. 2.11 3LGM² representation of domain layer, logical tool layer, and physical tool layer and their relationships of the function *patient admission* in a hospital

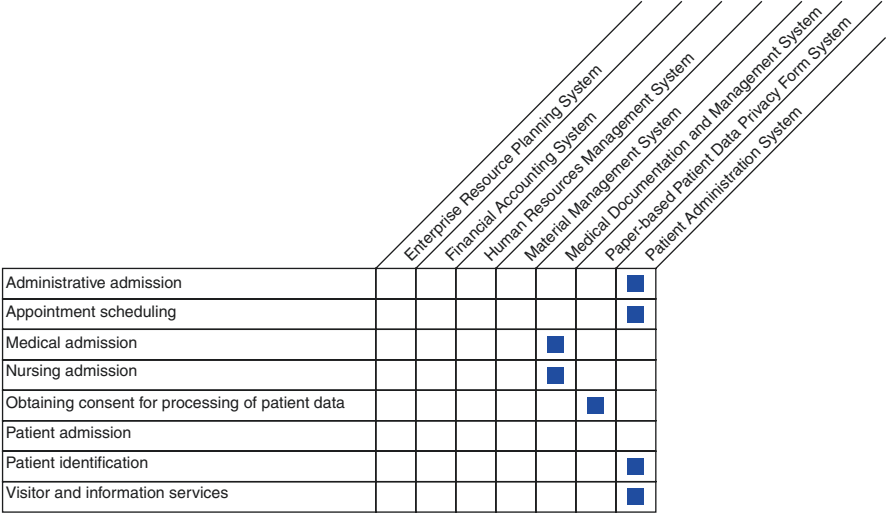


Fig. 2.12 The matrix visualizes the relations between functions and application components

The matrix view presented in Fig. 2.12 is an alternative representation of configuration lines between functions at the domain layer and application components at the logical tool layer (compare Fig. 2.11). Matrix views are also available for visualizing relations between other pairs of connected 3LGM² classes.

2.16 Exercises

2.16.1 Data, Information, and Knowledge

Imagine that a physician is given the following information about his patient, Mr. Russo: “Diagnosis: hypertension. Last blood pressure measurement: 160/100 mmHg.” Use this example to discuss the difference between “data,” “information,” and “knowledge”!

2.16.2 Systems and Subsystems

Look up some information on the nervous system of the human body. Then try to identify subsystems of the nervous system. In the same way, can you also describe subsystems of the system “hospital”?

2.16.3 Information Logistics

Imagine a situation in which a physician speaks with Mr. Russo at the patient's bedside. The physician looks up Mr. Russo's recent blood pressure measurement and ongoing medication, decides to increase the level of one medication, and explains this to Mr. Russo. Use this example to discuss the meaning of "information and knowledge logistics." What in this example indicates the right information, the right place, the right people, the right form, and the right decision? What could happen if an information system does not support high-quality information and knowledge logistics?

2.16.4 3LGM² Metamodel

Look at the 3LGM² example in Sect. 2.15. Use this example to explain the meaning of the following elements: functions, entity types, application systems, non-computer-based application components, physical data processing system, and inter-layer relationships.

2.16.5 Interpreting 3LGM² Models

Look at the 3LGM² sample model in Sect. 2.15 and try to answer the following questions.

- (a) Find examples of specialization or decomposition at the domain layer in Fig. 2.11.
- (b) What is the meaning of the arrows pointing from *patient identification* to "patient" and from "patient" to *medical admission* in Fig. 2.11?
- (c) What entity type that is stored in the paper-based patient data privacy form system should be added at the domain layer in Fig. 2.11?
- (d) Why is the function *patient admission* not connected with any application system in Fig. 2.11? (Hint: Look at the graphical representation of the domain layer in Fig. 2.11 and remember the modeling rules from Sect. 2.14.5.)
- (e) Which physical data processing systems are needed for the function "obtaining patient consent for the processing of data"?

References

1. World Health Organization (WHO) Health Promotion Glossary. WHO/HPR/HEP/98.1. 1998. <https://www.who.int/publications/i/item/WHO-HPR-HEP-98.1>. Accessed 15 Jan 2023.
2. Institute of Electrical and Electronics Engineers (2000). IEEE-Std 1471-2000: recommended practice for architectural description of software-intensive systems. <http://standards.ieee.org>. Accessed 15 Jan 2023.
3. Weill P, Ross JW. IT Governance. How top performers manage IT decision rights for superior results. Harvard: Harvard Business School Press; 2004. p. 34ff.

Open Access This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.

