

Multidimensional Data Modeling for Business Process Analysis

Svetlana Mansmann¹, Thomas Neumuth², and Marc H. Scholl¹

¹ University of Konstanz, P.O.Box D188, 78457 Konstanz, Germany
{Svetlana.Mansmann,Marc.Scholl}@uni-konstanz.de

² University of Leipzig, Innovation Center Computer Assisted Surgery (ICCAS),
Philipp-Rosenthal-Str. 55, 04103 Leipzig, Germany
Thomas.Neumuth@medizin.uni-leipzig.de

Abstract. The emerging area of business process intelligence attempts to enhance the analytical capabilities of business process management systems by employing data warehousing and mining technologies. This paper presents an approach to re-engineering the business process modeling in conformity with the multidimensional data model. Since the business process and the multidimensional model are driven by rather different objectives and assumptions, there is no straightforward solution to converging these models.

Our case study is concerned with Surgical Process Modeling which is a new and promising subdomain of business process modeling. We formulate the requirements of an adequate multidimensional presentation of process data, introduce the necessary model extensions and propose the structure of the data cubes resulting from applying vertical decomposition into flow objects, such as events and activities, and from the dimensional decomposition according to the factual perspectives, such as function, organization, and operation. The feasibility of the presented approach is exemplified by demonstrating how the resulting multidimensional views of surgical workflows enable various perspectives on the data and build a basis for supporting a wide range of analytical queries of virtually arbitrary complexity.

1 Introduction

Conventional business process management systems, focused on operational design and performance optimization, display rather limited analysis capabilities to quantify performance against specific metrics [1]. Deficiencies of business process modeling (BPM) approaches in terms of supporting comprehensive analysis and exploration of process data have been recognized by researchers and practitioners [1,2]. The new field of *Business Process Intelligence* (BPI), defined as the application of performance-driven management techniques from Business Intelligence (BI) to business processes, claims that the developing convergence of BI and BPM technologies will create value beyond the sum of their parts [3]. However, no straightforward guidelines for converging the flow-oriented process specification and the snapshot-based multidimensional design are in existence.

To be admitted into an OLAP (On-line Analytical Processing) system, the descriptions of the business processes have to undergo the transformation imposed by the underlying multidimensional data model. However, the source and the target models are driven by rather conflicting and partially incompatible objectives: business process modeling is concerned with operational efficiency and workflow behavior, whereas OLAP enables aggregation over accumulated numerical data modeled as a set of uniformly structured fact entries.

In medical engineering “the term *Surgical Workflows* refers to the general methodological concept of the acquisition of process descriptions from surgical interventions, the clinical and technical analysis of them” [4]. One of the major challenges is the acquisition of accurate and meaningful Surgical Process Models (SPM). Surgical Process Models are “simplified pattern of a surgical procedure that reflect a predefined subset of interest of the real intervention in a formal or semi-formal representation” [5]. Formalization of the SPM recording scheme is required to support both, manual and automatic data acquisition, and to apply state-of-the-art analysis and visualization techniques for gaining insight into the data.

Use cases of Surgical Workflows are manifold, ranging from supporting the preoperative planning by retrieving similar precedent cases to the postoperative exploration of surgical data, from analyzing the optimization potential with respect to instruments and systems involved to verifying medical hypotheses, for education purposes, answering qualitative and quantitative queries, etc. Whatever abstraction approach is adopted, there is a need for an unambiguous description of concepts that characterize a surgical process in a way adequate for modeling a wide range of different workflow types and surgical disciplines.

The prevailing process modeling standards, such as Business Process Modeling Notation (BPMN) [6] and the reference model of Workflow Management Coalition (WfMC) [7], are too general to address the domain-specific requirements adequately. Multidimensional modeling seems a promising solution as it allows to view data from different perspectives and at different granularity and define various measures of interest. To identify the major design challenges, we proceed by inspecting the fundamentals of the involved modeling techniques.

1.1 Multidimensional Data Model

Multidimensional data model emerged as an alternative to the relational data model optimized for quantitative data analysis. This model categorizes the data as *facts* with associated numerical *measures* and descriptive *dimensions* characterizing the facts [8]. Facts can thus be viewed as if shaped into a multidimensional cube with dimensions as axes and measure values as the cube cells. For instance, a surgical process can be modeled as a fact entry *SURGERY* characterized by dimensions *Location*, *Surgeon*, *Patient*, and *Discipline*. Members of a dimension are typically organized in a containment type hierarchy (e.g., *location* \nearrow *hospital* \nearrow *city*) to support multiple granularities.

Relational OLAP structures the data cubes according to the *star* or *snowflake* schema [9]. Both schemas are composed of a *fact table* and the associated

dimension tables. In the star schema, for each dimension, its whole hierarchy is placed into a single table, whereas the snowflake schema extracts each hierarchy level into a separate table and uses foreign keys for mapping child-parent relationships between the members. Within a dimension, the attributes that form the hierarchy are called *dimension levels*, or *categories*. Other descriptive attributes belonging to a particular category are *property attributes*. For instance, *hospital* and *city* are categories of the dimension *location*, whereas *hospital name* and *city code* are property attributes of the respective categories. Dimension levels along with parent-child relationships between them are referred to as the *intension*, or *schema*, of a dimension whereas the hierarchy of its members, i.e., the actual data tree, forms its *extension*.

1.2 Business Process Modeling and Workflow Management

BPM and Workflow Management (WfM) foster a process-oriented perspective on organizations that comprises *activities* and their *relationships* within and beyond an organization context. Relationships may be specified using control flow (consecutive, parallel, or alternative execution) and/or hierarchical decomposition; the organizational context comprises organizational units and resources [10]. The differentiation in the definition of business processes vs. workflows lies in the levels of abstraction: while business processes are mostly modeled in a high-level and informal way, workflow specifications serve as a basis for the largely automated execution and are derived by refining the business process specification [11]. A workflow is specified in terms of work steps, denoted *activities*, which are either automated or include a human part. The latter type is assigned roles filled by human actors at runtime. The role of the WfM system is to determine the (partial) invocation order of activities. Therefore, a formal specification of *control flow* and *data flow* is required.

Coexistence of different workflow specification methods is common in practice. We restrain ourselves to naming a few techniques applicable in the context of Surgical Workflows and refer the interested reader to [12] for a detailed overview. *Net-based*, or *graph-based*, methods enjoy great popularity due to their ability to visualize processes in a way understandable even for non-expert users. Especially the *activity and state charts* are frequently used to specify a process as an oriented graph with nodes representing the activities and arcs defining the ordering in which these are performed. *Logic-based* methods use temporal logic to capture the dynamics of the system. Finally, *Event-Condition-Action* rules are used for specifying the control flow between activities in the conditional form.

Surgical Process Modeling, classified as a specific domain of BPM [4], adopts the concepts from both WfM and BPM. The WfM approach of decomposing a workflow into activities is useful for providing a task-oriented surgery perspective. However, since surgical work steps are predominantly manual and involve extensive organizational context, such as participants, their roles, patients and treated structures, instruments, devices and other resources, etc., high-level BPM abstractions enable modeling such domain-specific elements.

2 Related Work

Relevant work can be subdivided into the following categories: 1) enhancing business process analysis by employing the data warehousing approach, 2) extending the OLAP technology to support complex scenarios, and 3) approaches to surgical workflow analysis.

Grigori et al. present a BPI tool suite built on top of the HP Process Manager (HPPM) and based on a data warehouse approach [2]. The process data is modeled according to the star schema, with process, service, and node state changes as facts and the related definitions as well as temporal and behavioral characteristics as dimensions. While this approach focuses on the analysis of process execution and state evolution, we pursue the task-driven decomposition into logical work steps, in which horizontal characteristics, or the factual perspectives [13], extended by means of domain-specific taxonomies serve as dimensions.

An approach to visual analysis of business process performance metrics, called impact factors, is given in [14]. The proposed visualization interface *VisImpact* is especially suitable for aggregating over large amounts of process-related data and is based on analyzing the process schema and instances to identify business metrics. The selected impact factors and the corresponding process instances are presented using a symmetric circular graph to display the relationships and the details of the process flows.

Pedersen et al. have made remarkable contributions in the field of multidimensional modeling for non-standard application domains. In [15], a medical cases study concerned with patient diagnosis is used to demonstrate the analysis requirements not supported by traditional OLAP systems. The proposed model extensions aim at supporting non-summarizable hierarchies, symmetric treatment of dimensions and measures, and correct aggregation over imprecise or incomplete data. In [16], Jensen et al. present the guidelines for designing complex dimensions in the context of spatial data such as mobile, location-based services.

In a previous work [17] we analyzed the limitations of conventional OLAP systems and the underlying data model in handling complex dimension hierarchies and proposed model extensions at the conceptual level and their relational mapping as well their implementation in a prototype frontend tool. A comprehensive classification of dimensional hierarchies, including those not addressed by current OLAP systems, formalized at both the conceptual model and the logical level, may be found in [18].

Interdisciplinary research in the field of surgical workflow modeling, analysis and visualization is carried out at the Innovation Center Computer Assisted Surgery (ICCAS) located in Leipzig, Germany. Recent results and findings of the ongoing projects may be found in [4,5].

3 Case Study: Surgical Workflows

Surgeons, medical researchers and engineers work jointly on obtaining a well-defined formal Surgical Process Model that would enable managing huge volumes

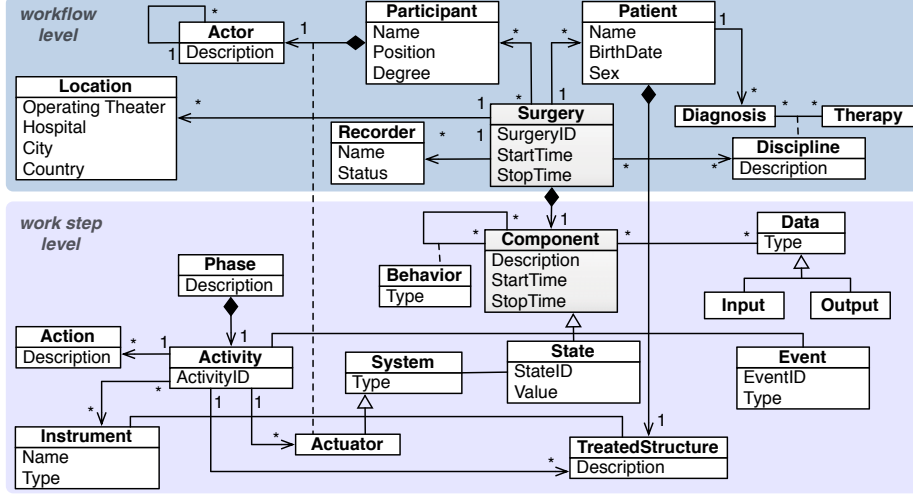


Fig. 1. Recording scheme of a surgical process model as a UML class diagram

of intervention models in a single data warehouse in a uniform manner and querying that data for analytical purposes. A basic recording scheme of a surgery in UML class notation is shown in Figure 1. The diagram denotes a further stage of the scheme presented by Neumuth et al. in [4]. The use of UML offers an implementation-independent view of the process scheme and is a widely accepted specification standard for both BPM [19] and data warehouse design [20]. The upper part of the diagram contains the characteristics describing the surgery as a whole and corresponding to the dimensions of analysis for aggregating across multiple surgical interventions (for instance, to query the number of patients treated by a particular surgeon). Classes in the lower part of the diagram belong to the intra-surgical level, i.e., they represent elements constituting a surgical procedure.

To obtain the structure of a workflow recording scheme whilst avoiding the information overload, we employ vertical and horizontal process decomposition.

Vertical decomposition corresponds to identifying core elements of a process. Here, we account for two complementary data acquisition practices in the field of SPM, namely a task-driven, or temporal, and an system-based structuring. *Activities* represent surgical tasks, or work steps, similarly to the corresponding WfM concept. Examples of activities are “irrigation of a vessel with a coagulator” or “cutting at the skin with a scalpel”. Sequential ordering of activities symbolizes the acquired surgical intervention [4]. System-based structuring uses the concepts of *System*, *State*, and *Event* to capture the state evolution of involved systems and events that trigger state transitions. The concept of a system is very generic and may refer to a participant or his/her body part, a patient or a treated structure, an instrument or a device, etc. For instance, the gaze direction of surgeon’s eyes can be modeled as states, while surgeon’s instructions may be captured as events. To reflect the heterogeneous nature of the notion *system*, we

modeled it as an abstract superclass as shown in Figure 1. Another superclass **Component** enables uniform treatment of the two data acquisition practices in part of their common properties, e.g., to retrieve the entire output generated in the course of a surgery, whether by its activities, system states or events.

Horizontal decomposition of a process is conceptually similar to identifying the dimensions of a data cube and is drawn by recognizing different complementary perspectives in a workflow model, following the factual perspective categorization [13]. Further details on each perspective are given in the next section.

4 From Process Flows to Data Cubes

Transformation from the semantically rich BPM notation into a data cube can be seen as a reduction of the complete set of extensible process elements, such as various types of flow and connecting objects, to a rigid format that forces decomposition into a set of uniformly structured facts with associated dimensions. We proceed in three steps: 1) identify the main objectives of the business process analysis, 2) provide the overall mapping of generic BPM concepts, such as activity, object, resource, event etc. into the multidimensional data model, and 3) transfer the application-specific characteristics into the target model.

Subjects, or focal points, of the analysis are mapped to facts. In business process analysis, the major subjects of the analysis are the process itself (process level) as well as its components (intra-process level). Process level analysis is concerned with analyzing the characteristics of the process as a whole and aggregating over multiple process instances. Back to our case study, sample analytical tasks at this level are the utilization of hospital locations, surgery distribution by discipline, surgeon ranking, etc. At the intra-process level, occurrence, behavior and characteristics of process components, such as activities, actors, and resources are analyzed. Examples from the surgical field are the usage of instrument and devices, work step duration, occurrence of alarm states, etc.

4.1 Handling Generic BPM Constructs

The conceptual design of a data warehouse evolves in modeling the structure of business facts and their associated dimensions. Once major fact types have been defined, aggregation hierarchies are imposed upon dimensions to enable additional granularities. In what follows we present a stepwise acquisition of the multidimensional perspective of a process.

Determining the Facts. As the fact entries within a data cube are required to be homogeneous, i.e., drawn from the same set of dimensions, applications dealing with multiple heterogeneous process types have to place each type into a separate cube. In our scenario, surgery is the only process type, but if we had to add a different type, e.g., a routine examination of a patient, the corresponding fact entries would be stored separately from surgical facts.

At the process element level, we suggest modeling work steps, or activities, as facts while other components, such as resources and actors, are treated as

dimensional characteristics of those facts. However, in many contexts, process activities may be rather heterogeneous in terms of their attributes. To preserve homogeneity within the fact type, we propose to extract each homogeneous group of activity types into a separate fact type. To account for common characteristics of all activity types, generalization into a common superclass is used.

Determining the Dimensions. Dimensions of a fact are a set of attributes determining the measure value of each fact entry. These attributes are obtained via a horizontal decomposition along the factual perspective categories of workflow modeling defined in [13]. Availability and contents of particular perspective categories as well as their number depend on the type of process at hand. Our approach to transforming the fundamental factual perspectives into dimensions is as follows:

1. The *function* perspective describes recursive decomposition of process into subprocesses and tasks. This composition hierarchy is mapped into a dimension of **Activity**, such as **Phase** in our case study.
2. The *operation* perspective describes which operations are supported by a task and which applications implement these operations. In case of a surgical work step, operations are mapped to the dimension **Action** (e.g., “cut”, “suction”, “stitch up”, etc.) and the applications are represented by **Instrument**.
3. The *behavior* perspective defines the execution order within the process. Behavior can be subdivided into temporal (along the timeline), logical (parallelism, synchronization, looping) and causal. Temporal characteristics, such as **StartTime** and **StopTime**, are used as time dimensions. Relationships between pairs of components (a reflexive association of **Component** with **Behavior** in Figure 1) are more complex and will be discussed in the next section.
4. The *information* perspective handles the data consumed and produced by the workflow components. These resources can be mapped to (**Input**) and (**Output**) dimensions.
5. The *organization* perspective specifies which resource is responsible which task. Organization dimensions may involve human actors, systems, and devices. Back to the surgical activity case, an example of such resource is **Participant** (e.g., “surgeon”, “assistant”, etc.).

5 Challenges of the Multidimensional Modeling

Apart from the standard OLAP constraints, such as normalization of the dimension hierarchies and avoidance of NULL values in the facts, the following domain-specific requirements have been identified:

- *Many-to-many relationships between facts and dimensions* are very common. For instance, during a single surgery, multiple surgical instruments are used by multiple participants.
- *Heterogeneity of fact entries.* Treating **Component** elements as the same fact type would disallow capturing of subclass specific properties, while modeling

each subclass as a separate fact type would disable treating heterogeneous elements as the same class for querying their common characteristics.

- *Interchangeability of measure and dimension roles.* In a classical OLAP scenario the measures of interest are known at design time. However, “raw” business process data may contain no explicit quantitative characteristics. The measure of interest varies from one query to another. Therefore, it is crucial to enable the runtime measure specification from virtually any attribute. For instance, a query may investigate the number of surgeries per surgeon or retrieve the distribution of surgeons by discipline.
- *Interchangeability of fact and dimension roles.* *Surgery* has dimensional characteristics of its own (location, patient, etc.) and therefore, deserves to be treated as a fact type. However, with respect to single work steps, *Surgery* clearly plays the role of a dimension (e.g., events may be rolled-up to surgery).

5.1 Terminology

In this work, we adopt the notation proposed by Pedersen et al. [15] by simplifying and extending it to account for BPM particularities.

An *n-dimensional fact schema* is a pair $\mathcal{S} = (\mathcal{F}, \{\mathcal{D}_i, i = 1, \dots, n\})$, with \mathcal{F} as the fact schema and $\{\mathcal{D}_i\}$ as the set of corresponding dimension schemata.

A *dimension schema* is a four-tuple $\mathcal{D} = (\{\mathcal{C}_j, j = 1, \dots, m\}, \sqsubseteq_{\mathcal{D}}, \top_{\mathcal{D}}, \perp_{\mathcal{D}})$, where $\{\mathcal{C}_j\}$ are the categories, or aggregation levels, in \mathcal{D} , with the distinguished top and bottom category denoted $\top_{\mathcal{D}}$ and $\perp_{\mathcal{D}}$, respectively, and $\sqsubseteq_{\mathcal{D}}$ being the partial order on the \mathcal{C}_j s.

The top category of a dimension corresponds to an abstract root node of the data hierarchy and has a single value referred to as **ALL** (i.e., $\top_{\mathcal{D}} = \{\text{ALL}\}$).

A non-top *dimension category* is a pair $\mathcal{C} = (\{\mathcal{A}_k, k = 1, \dots, p\}, \bar{\mathcal{A}}_{\mathcal{C}})$ where $\bar{\mathcal{A}}_{\mathcal{C}}$ is the distinguished *hierarchy attribute*, i.e., whose values represent a level in the dimension hierarchy, whereas $\{\mathcal{A}_k\}$ is a set of *property attributes* functionally dependent on $\bar{\mathcal{A}}_{\mathcal{C}}$, i.e., $\forall \mathcal{A}_k \in \mathcal{C} : \mathcal{A}_k = f(\bar{\mathcal{A}}_{\mathcal{C}})$.

A *fact schema* is a triple $\mathcal{F} = (\{\bar{\mathcal{A}}_{\perp}\}_{\mathcal{F}}, \{\mathcal{M}_q, q = 1, \dots, t\}, \bar{\mathcal{A}}_{\mathcal{F}})$, where $\{\bar{\mathcal{A}}_{\perp}\}$ is a set of bottom-level hierarchy attributes in the corresponding dimension schema $\{\mathcal{D}_i\}$ (i.e., $\forall \mathcal{C} = \perp_{\mathcal{D}_i} : \bar{\mathcal{A}}_{\mathcal{C}} \in \{\bar{\mathcal{A}}_{\perp}\}_{\mathcal{F}}$), $\{\mathcal{M}_q\}$ is a set of *measure attributes*, defined by its associated dimensions, such that $\forall \mathcal{M}_q \in \mathcal{F} : \mathcal{M}_q = f(\{\bar{\mathcal{A}}_{\perp}\}_{\mathcal{F}})$, and $\bar{\mathcal{A}}_{\mathcal{F}}$ is an optional *fact identifier attribute*.

We allow the set of measure attributes to be empty ($\{\mathcal{M}_q\} = \emptyset$), in which case the resulting fact schema is called *factless* [9] and the measures need to be defined dynamically by applying the desired aggregation function to any category in $\{\mathcal{D}_i\}$. The fact identifier attribute plays the role of a single-valued primary key, useful for specifying the relationship between different fact schemata.

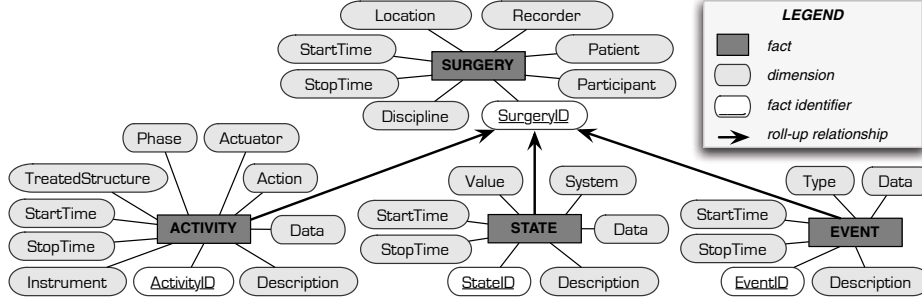


Fig. 2. Vertical decomposition of the surgical workflow into a fact hierarchy

5.2 Fact Constellation vs. Fact Hierarchy and Fact Generalization

In our usage scenario, fact table modeling is an iterative process starting with a coarse definition of the basic fact types with their subsequent refinement under the imposed constraints. Vertical decomposition of a surgical process results in two granularity levels of the facts, as depicted in Figure 2:

- **Surgery.** Each surgical case along with its attributes and dimensional characteristics represents the top-level fact type.
- **Activity, State, and Event.** The three types of workflow components have their specific sets of dimensions and are thus treated as distinct fact types.

At this initial stage, we disregarded existence of many-to-many relationships between facts and dimensions. However, disallowance of such relationships is crucial in the relational context as each fact entry is stored as a single data tuple with one single-valued attribute per dimension. Consider the problem of modeling **Participant** as a dimension of **Surgery**: most surgeries involve multiple participants, hence, it is impossible to store the latter as a single-valued attribute.

Our solution is based on a popular relational implementation of a non-strict dimension hierarchy by means of *bridge tables* [9]. A bridge table captures a non-strict ordering between any two categories by storing each parent-child pair. Back to our example, a many-to-many relationship between **Surgery** and **Participant** as well as that between **Surgery** and **Discipline** are extracted each into a separate table, as shown in Figure 3. We denote such extracted fact-dimensional fragments *satellite facts* to stress their dependent nature. Availability of the fact identifier attribute **SurgeryID** facilitates the connection of the satellite fact to its base fact

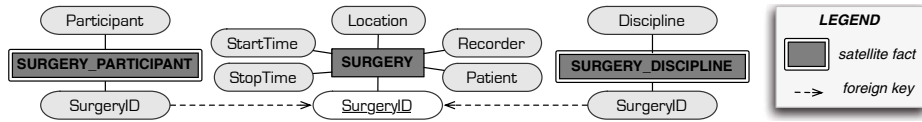


Fig. 3. Extracting many-to-many relationships into “satellite” facts

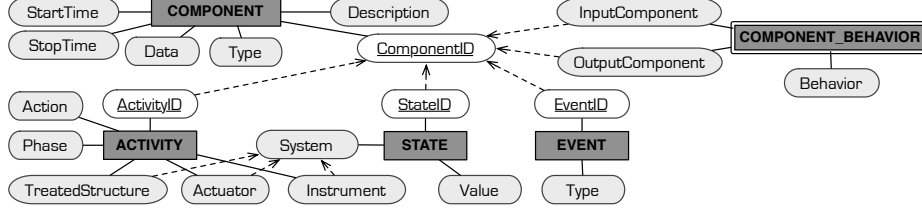


Fig. 4. Using generalization (dashed lines) for unifying heterogeneous categories

table; a natural join between the two fact tables is necessary in order to obtain the entire multidimensional view of *Surgery*.

Another phenomenon worthwhile consideration is the presence of parent-child relationships between fact types, such as the hierarchy *Activity* \nearrow *Surgery*. Similar to a hierarchical dimension, *Activity* records can be rolled-up to *Surgery*.

A *fact hierarchy* relationship between \mathcal{F}_j and \mathcal{F}_i , denoted $\mathcal{F}_j \nearrow \mathcal{F}_i$, is a special case of the fact constellation in which the fact schema \mathcal{F}_i appears to serve as a dimension in \mathcal{F}_j , such that $\bar{\mathcal{A}}_{\mathcal{F}_i} \in \{\bar{\mathcal{A}}_{\perp}\}_{\mathcal{F}_j}$.

So far, the three workflow component types have been modeled as separate fact types *Activity*, *State*, and *Event*. However, these heterogeneous classes have a subset of common characteristics that qualify them to be generalized into superclass fact type *Component*, resulting in a *fact generalization* depicted in Figure 4. A simple relational implementation of *Component* can be realized by defining a corresponding view as a union of all subclass projections onto the common subset of schema attributes.

\mathcal{F}_j is a *fact generalization* of \mathcal{F}_i , denoted $\mathcal{F}_j \subset \mathcal{F}_i$, if the dimension and measure sets of \mathcal{F}_j are a subset of the respective sets in \mathcal{F}_i :

$$\{\bar{\mathcal{A}}_{\perp}\}_{\mathcal{F}_j} \subset \{\bar{\mathcal{A}}_{\perp}\}_{\mathcal{F}_i} \wedge (\forall \mathcal{M}_q \in \mathcal{F}_j : \mathcal{M}_q \in \mathcal{F}_i).$$

An obvious advantage of the generalization is the ability to treat heterogeneous classes uniformly in part of their common characteristics. A further advantage is the ability to model the behavior of components with respect to each other (see *Behavior* class in Figure 1) in form of a satellite fact table *Component_Behavior* depicted in Figure 4.

5.3 Modeling Dimension Hierarchies

A key strategy in designing dimension hierarchies for OLAP is that of *summarizability*, i.e., the ability of a simple aggregate query to correctly compute a higher-level cube view from a set of precomputed views defined at lower aggregation levels. Summarizability is equivalent to ensuring that 1) facts map directly to the lowest-level dimension values and to only one value per dimension, and 2) dimensional hierarchies are balanced trees [21]. Originally motivated by performance considerations, the summarizability has regained importance in the

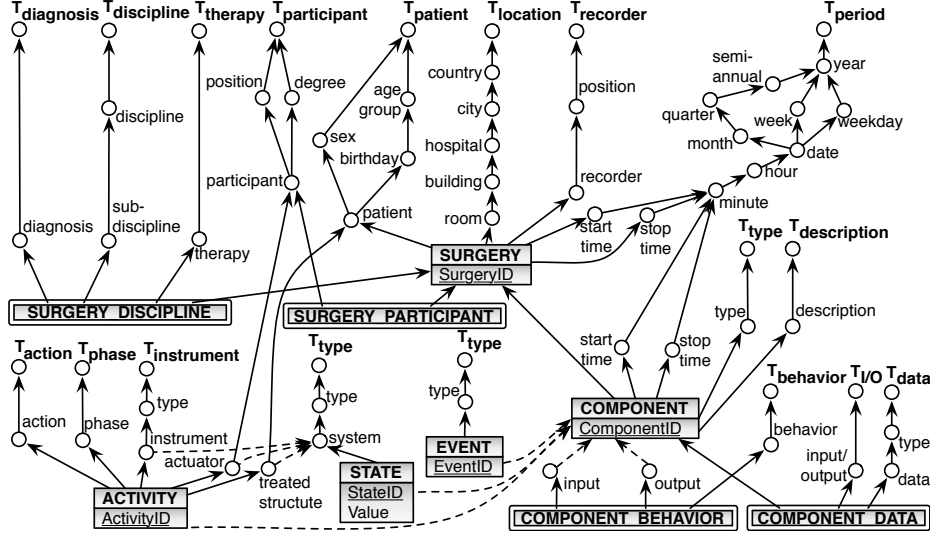


Fig. 5. A (simplified) Dimensional Fact Model of a surgical workflow scheme

context of visual OLAP as it ensures the generation of a proper browser-like navigation for visual exploration of multidimensional cubes [17].

The resulting structure of the entire surgery scheme (with some simplifications) in terms of facts, dimension hierarchies, and the relationships between them is presented in Figure 5 in the notation similar to the Dimensional Fact Model [22]. Solid arrows show the roll-up relationships while dashed arrows express the “is a” relationships, namely the identity in case of a satellite fact and the generalization in case of a fact hierarchy. The chosen notation is helpful for explicitly presenting all shared categories, and therefore, all connections and valid aggregation paths in the entire model.

We limit ourselves to naming a few non-trivial cases of dimensional modeling.

Multiple alternative hierarchies. The time hierarchy in the dimension **Period** is a classical example of alternative aggregation paths, such as **date** \nearrow **month** and **date** \nearrow **week**. These paths are mutually exclusive, i.e., within the same query, the aggregates may be computed only along one of the alternative paths.

Parallel hierarchies in a dimension account for different analysis criteria, for example, the member values of **Patient** can be analyzed by age or by sex criteria. Apparently, such hierarchies are mutually non-exclusive, i.e., it is possible to compute the aggregates grouped by age and then by sex, or vice versa.

Generalization hierarchies are used to combine heterogeneous categories into a single dimension. **System** is an example of a superclass, which allows to model the belonging of the categories **Instrument**, **TreatedStructure**, and **Actuator** to the dimension **System** of the fact type **STATE**, as shown in Figure 4.

Fact as dimension. In the case of a fact hierarchy or a satellite fact, the whole n -dimensional fact schema \mathcal{S} of the basis fact is included as a hierarchical dimension into its dependent fact. For instance, COMPONENT treats SURGERY as its dimension, while the dimensions Patient, Location, etc. of the latter are treated as parallel hierarchies [18] within the same dimension.

Dimension inclusion is a special case of shared dimensions, in which dimension \mathcal{D}_j represents a finer granularity of dimension \mathcal{D}_i , or formally, $\mathcal{D}_i \subset \mathcal{D}_j$ if $\exists \mathcal{C}_k \in \mathcal{D}_j : \mathcal{C}_k \sqsubset \perp_{\mathcal{D}_i}$. For example, TreatedStructure in ACTIVITY rolls up to Patient in SURGERY. Dimension inclusion implies that all categories in \mathcal{D}_i become valid aggregation levels of \mathcal{D}_j .

The guidelines for modeling complex dimensions are provided in [15,18,17].

5.4 Runtime Measure Specification

Compulsory elements of any aggregate query are 1) a measure specified as an aggregate function (e.g., sum, average, maximum etc.) and its input attribute, and 2) a set of dimension categories to use as the granularity of the aggregation. Conventional OLAP tools require the set of the available measures within a cube to be pre-configured at the metadata level. It is also common to provide a wizard for defining a new measure, however, limiting the selection of qualifying attributes to the set \mathcal{M}_q of fact schema \mathcal{F} , i.e., to the actual measure attributes encountered in the fact table. In our scenario, the measure definition routine needs to be modified to account for the following phenomena:

- The fact schema is factless, i.e., $\{\mathcal{M}_q\} = \emptyset$.
- Each non-satellite fact schema disposes of a fact identifier attribute $\bar{\mathcal{A}}_{\mathcal{F}}$ belonging neither to the measure nor to the dimension set of \mathcal{F} .
- Any attribute of a data cube, whether of the fact table itself or of any of its dimensions, can be chosen as an input for a measure. Examples of commonly queried measures are the total number of patients operated, average number of surgeries in a hospital, most frequent diagnoses, number of distinct instruments per surgery, etc.

In accordance with the above requirements, we propose to enable runtime measure specification by the analyst as a 3-step process, depicted in Figure 6:

1. Selecting an aggregate function from the function list;
2. Specifying the measure attribute: in a visual interface, this can be done via a “drag&drop” of a category from the navigation, as shown in Figure 6, where Hospital category is being dragged into the measure window;
3. Specifying whether the duplicates should be eliminated from the aggregation by activating the DISTINCT option.

Define new measure	
Name	Number of participants
Function	SUM
Attribute	Drag any category in here Hospital
<input checked="" type="checkbox"/> DISTINCT	
<div>Cancel</div> <div>OK</div>	

Fig. 6. Defining a measure

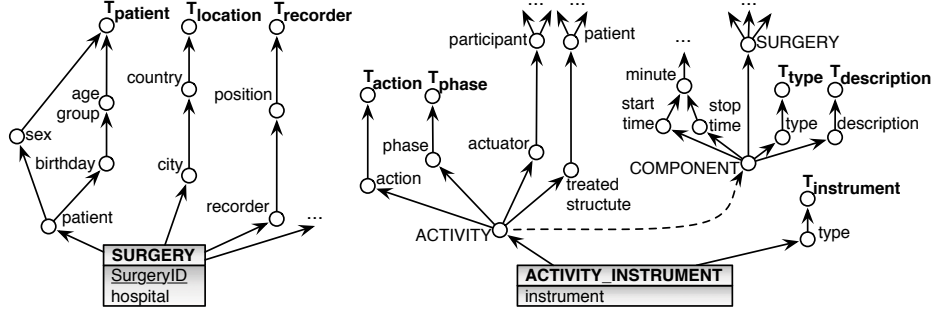


Fig. 7. Changes in the conceptual schema caused by deriving a measure from a dimension category: (*left*) number of hospitals, (*right*) number of instruments

Optionally, the newly defined measure may be supplied with a user-friendly name. As long as no user-defined measure is specified, the default setting of `COUNT(*)`, i.e., simple counting of the qualifying fact entries, is used. In terms of the conceptual model, derivation of a measure from virtually any element of the n -dimensional fact schema is equivalent to re-designing the entire schema.

Let us consider an example of analyzing the number of hospitals, i.e., using category `Hospital` from dimension `Location` as the measure attribute. Obviously, to support this measure, `SURGERY` facts need to be aggregated to the `Hospital` level, `Hospital` turns into a measure attribute within `SURGERY` and the bottom granularity of `Location` changes from `Room` to `City`. The resulting data schema is shown in Figure 7 (left). `Location` granularities below `Hospital` simply become invalid in the defined query context.

A more complicated example of selecting the number of instruments to serve as a measure is presented in Figure 7 (right). `Instrument` category is turned into a measure attribute of the fact table `ACTIVITY_INSTRUMENT`. From this perspective, all upper-level facts, such as `ACTIVITY` and `SURGERY`, are treated as dimension categories. Thus, the analyst may pursue any aggregation path valid in the context of the chosen measure. For example the number of instruments can be rolled-up to `SURGERY`, `Action`, `Phase`, etc.

In practice, the schemata of the designed data cubes remains unchanged and only a virtual view corresponding to the adjusted schema is generated to support querying user-defined measures. For frequently used measures, materialization of the respective view may improve the performance.

6 Results

The feasibility of our model can be shown by implementing it into a relational OLAP system and running domain-specific queries against the accumulated data. We present an application case of analyzing the use of instruments in the surgical intervention type discectomy. The goal of a discectomy is partial

	Measures							
	COUNT(ActivityID)				AVG(StopTime – StartTime)			
	SurgeryID							
Dimensions								
Instrument	A	B	C	D	A	B	C	D
coagulator	7	17	4	8	00:00:31	00:00:23	00:00:34	00:00:27
dissector	3	3	14	4	00:00:56	00:00:16	00:00:25	00:00:45
forceps	12	3	7	10	00:01:50	00:00:32	00:00:54	00:01:51
hook	15	7	7	12	00:01:14	00:01:01	00:00:31	00:00:47
punch	9	22	10	9	00:02:38	00:00:35	00:00:46	00:01:27
scalpel	2	3	2	2	00:00:53	00:01:23	00:00:22	00:01:09
suction tube	6	26	2	2	00:14:42	00:00:12	00:16:29	00:11:21
Total	54	81	46	47	00:03:15	00:00:37	00:02:52	00:02:32

Fig. 8. Results of sample aggregate queries 1 und 2 as a pivot table

removal of the herniated intervertebral disc. Typical expert queries in this scenario focus on the occurrence of particular instruments, frequency of their usage throughout the surgery, and duration of usage periods. Figure 8 shows a pivot table with the results of the following two queries:

Query 1. *For each of the interventions of type discectomy, find the instruments used by the surgeon and the frequency of their occurrence (i.e., the number of activities in which that instrument is used).*

The measure of this query, i.e., the number of activities (COUNT(DISTINCT ActivityID)), is rolled-up by SurgeryID and Instrument with a selection condition along Discipline. The input data cube is obtained by joining the fact tables SURGERY and ACTIVITY with their respective satellites SURGERY_DISCIPLINE and ACTIVITY_INSTRUMENT and joining the former two with each other via COMPONENT. The left-hand half of the table in Figure 8 contains the computed occurrence aggregates, with Instrument mapped to the table rows and SurgeryID as well as the measure COUNT(DISTINCT ActivityID) in the columns.

Query 2. *For each of the interventions of type discectomy, calculate the mean usage times of each instrument used by the surgeon (i.e., the average duration of the respective activities).*

The duration of a step corresponds to the time elapsed between its start and end, so that the measure can be specified as (AVG(StopTime–StartTime)). The rollup and the filtering conditions are identical to the previous query. The resulting aggregates are contained in the right-hand half of the pivot table.

Other examples of surgical queries supported by our proposed multidimensional design for Surgical Workflows are ‘How much time does the surgeon spend on action X?’, ‘At which anatomical structures has instrument Y been used?’, or ‘Which input is needed to execute a particular work step?’.

7 Conclusion

In this work we applied the data warehousing approach to business process analysis. Conventional BPMS are rather limited in the types of supported analysis tasks, whereas data warehousing appears more suitable when it comes to managing large amounts of data, defining various business metrics, and running complex queries. The case study presented in this work is concerned with designing a recording scheme for acquiring process descriptions from surgical interventions for their subsequent analysis and exploration.

As the business process model and the multidimensional model are based on different concepts, it is crucial to find a common abstraction for their convergence. We propose to map the vertical decomposition of a process into temporal or logical components to fact entries at two granularity levels, namely, at the process and at the work step level. Horizontal decomposition according to the factual perspectives, such as function, organization, operation, etc., is used to identify dimensional characteristics of the facts.

We evaluated the relational OLAP approach against the requirements of our case study and proposed an extended data model that addresses such challenges as non-quantitative and heterogeneous facts, many-to-many relationships between facts and dimensions, runtime definition of measures, interchangeability of fact and dimension roles, etc. The proposed model extensions can be easily implemented using current OLAP tools, with facts and dimensions stored in relational tables and queried with standard SQL. We presented a prototype of a visual interface for the runtime measure definition and concluded the work by producing the results of sample analytical queries formulated by the domain experts and run against the modeled surgical process data warehouse.

Acknowledgement

We would like to thank Oliver Burgert from ICCAS at the University of Leipzig as well as Christos Trantakis and Jürgen Meixensberger from the Neurosurgery Department at the University Hospital of Leipzig for their expert support.

References

1. Dayal, U., Hsu, M., Ladin, R.: Business process coordination: State of the art, trends, and open issues. In: VLDB 2001: Proc. 27th Int.Conf. on Very Large Data Bases, pp. 3–13 (2001)
2. Grigori, D., Casati, F., Castellanos, M., Dayal, U., Sayal, M., Shan, M.-C.: Business process intelligence. *Computers in Industry* 53(3), 321–343 (2004)
3. Smith, M.: Business process intelligence. *Intelligent Enterprise*, Online (December 2002), http://www.intelligententerprise.com/021205/601feat2_1.jhtml
4. Neumuth, T., Strauß, G., Meixensberger, J., Lemke, H.U., Burgert, O.: Acquisition of process descriptions from surgical interventions. In: Bressan, S., Küng, J., Wagner, R. (eds.) DEXA 2006. LNCS, vol. 4080, pp. 602–611. Springer, Heidelberg (2006)

5. Neumuth, T., Trantakis, C., Eckhardt, F., Dengl, M.: Supporting the analysis of intervention courses with surgical process models on the example of fourteen microsurgical lumbar discectomies. *International Journal of Computer Assisted Radiology and Surgery* 2(1), 436–438 (2007)
6. OMG (Object Management Group): BPMN (Business Process Modeling Notation) 1.0: OMG Final Adopted Specification, Online (February 2006), <http://www.bpmn.org>
7. WfMC (Workflow Management Coalition): WfMC Standards: The Workflow Reference Model, Version 1.1, Online (January 1995), <http://www.wfmc.org/standards/docs/tc003v11.pdf>
8. Pedersen, T.B., Jensen, C.S.: Multidimensional database technology. *IEEE Computer* 34(12), 40–46 (2001)
9. Kimball, R., Reeves, L., Ross, M., Thornthwaite, W.: *The Data Warehouse Lifecycle Toolkit*. John Wiley & Sons, Inc., New York (1998)
10. Jung, J.: Meta-modelling support for a general process modelling tool. In: *DSM 2005: Proc. 5th OOPSLA Workshop on Domain-Specific Modeling*, pp. 602–611 (2005)
11. Muth, P., Wodtke, D., Weißenfels, J., Weikum, G., Kotz-Dittrich, A.: Enterprise-wide workflow management based on state and activity charts. In: *Proc. NATO Advanced Study Institute on Workflow Management Systems and Interoperability*, pp. 281–303 (1997)
12. Matousek, P.: *Verification of Business Process Models*. PhD thesis, Technical University of Ostrava (2003)
13. Jablonski, S., Bussler, C.: *Workflow Management. Modeling Concepts, Architecture and Implementation*. International Thomson Computer Press (1996)
14. Hao, M.C., Keim, D.A., Dayal, U.: Business process impact visualization and anomaly detection. *Information Visualization* 5, 15–27 (2006)
15. Pedersen, T.B., Jensen, C.S., Dyreson, C.E.: A foundation for capturing and querying complex multidimensional data. *Information Systems* 26(5), 383–423 (2001)
16. Jensen, C.S., Kligys, A., Pedersen, T.B., Timko, I.: Multidimensional data modeling for location-based services. *The VLDB Journal* 13(1), 1–21 (2004)
17. Mansmann, S., Scholl, M.H.: Empowering the OLAP technology to support complex dimension hierarchies. *International Journal of Data Warehousing and Mining* 3(4), 31–50 (2007)
18. Malinowski, E., Zimányi, E.: Hierarchies in a multidimensional model: From conceptual modeling to logical representation. *Data & Knowledge Engineering* 59(2), 348–377 (2006)
19. Hruby, P.: Structuring specification of business systems with UML (with an emphasis on workflow management systems). In: *Proc. OOPSLA'98 Business Object Workshop IV*, Springer, Heidelberg (1998)
20. Luján-Mora, S., Trujillo, J., Vassiliadis, P.: Advantages of uml for multidimensional modeling. In: *ICEIS 2004: Proc. 6th Int. Conf. on Enterprise Information Systems*, pp. 298–305 (2004)
21. Lenz, H.-J., Shoshani, A.: Summarizability in OLAP and statistical data bases. In: *SSDBM 1997: Proc. of 9th Int. Conf. on Scientific and Statistical Database Management*, pp. 132–143 (1997)
22. Golfarelli, M., Maio, D., Rizzi, S.: The dimensional fact model: A conceptual model for data warehouses. *International Journal of Cooperative Information Systems* 7(2-3), 215–247 (1998)