Process Mining: Multi Dimensional Cubes

[1]S. Sowjanya. Chintalapati, [2]J. Sowjanya, [3]T. Charan Singh, [4]A. Rangamma
[1,2,3,4.] Asst Professors,
Computer Science and Engg, SRI INDU College of Engg and Technology, Ibrahimpatnam, Hyderabad, India

***ABSTRACT*--Business process analysis ranges from model verification at design-time to the monitoring of processes at runtime. Much progress has been achieved in process verification. Recent breakthroughs in process mining research make it possible to discover, analyze, and improve business processes based on event data. The growth of event data provides many opportunities but also imposes new challenges. Process mining is typically done for an isolated well-defined process in steady-state.**

Process mining tools have in common is that installation and maintenance of the systems requires enormous effort, and deep knowledge of the underlying information system. This paper proposes the notion of process cubes where events and process models are organized using different dimensions. Each cell in the process cube corresponds to a set of events and can be used to discover a process model, to check conformance with respect to some process model, or to discover bottlenecks. The idea is related to the well-known OLAP [2] (Online Analytical Processing) data cubes and associated operations such as slice, dice, roll-up, and drill-down.

Keywords:* *Process Mining, Big Data, Process Discovery, OLAP, MDA.

I. INTRODUCTION: THE ROLE OF MODELS

Models play an important role in information systems and it is clear that the importance of models will increase. Models can be used to specify systems and processes and can be used for their analysis. Some of today's information systems are even driven by models (cf. workflow management systems).

Although the general vision of a "Model Driven Architecture"(MDA) is appealing, it is not yet realistic/practical for many applications. Only in specific niches such as workflow technology, MDA is already a reality and has proven to be valuable. Although the general vision of a "Model Driven Architecture" (MDA) (Refer Fig 1) is a software design approach for the development of software systems. It provides a set of guidelines for the structuring of specifications, which are expressed as models. MDA tool is a tool used to develop, interpret, compare, align, measure, verify, transform, etc. models. A "model" is interpreted as any kind of model (e.g. UML is a model). In any MDA approach we have essentially two kinds of models: initial models are created manually by human agents while derived models are created automatically by programs [10]. For example an analyst may create a UML initial model from its observation of some loose business situation while a Java model may be automatically derived from this UML model by a Model transformation operation**.**

Fig 1) Model Driven Architecture (MDA)

II. PROCESS MINING

Many organizations realize that increasing amounts of \Big Data" (in the broadest sense of the word) need to be used intelligently in order to compete with other organizations in terms of efficiency, speed and service. However, the goal is not to collect as much data as possible. The real challenge is to turn event data into valuable insights. Only process mining techniques directly relate event data to end-to-end business processes [1]. Existing business process modeling approaches generating piles of process models are typically disconnected from the real processes and information systems. Data oriented analysis techniques (e.g., data mining and machines learning) typically focus on simple classification, clustering, regression, or rule-learning problems.

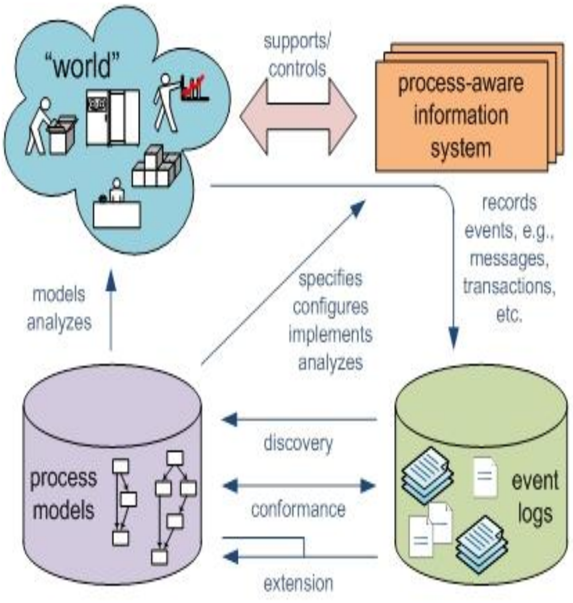

"world"
supports/ controls
process-aware information system
records events, e.g., messages, transactions, etc.
models analyzes
specifies configures implements analyzes
process models
discovery
conformance
extension
event logs

Fig 2) Traditional Model for Process mining.

Process mining techniques can be used to extract knowledge from event data, discover models, align logs and models, measure conformance, diagnose bottlenecks, and predict future events. Today's processes leave many trails in data bases, audit trails, message logs, transaction logs, etc. (Refer Fig 2) Therefore; it makes sense to relate these event data to process models independent of their particular notation. [7]Process models discovered based on the actual behavior tend to be very different from the process models made by humans. Moreover, conformance checking techniques often reveal important deviations between models and reality.

Traditionally, process models and system specifications tend to be static and disconnected from the real processes and system. Process mining techniques provide a means to establish a direct connection between processes, models, and systems. Moreover, event data can be used to breathe life into process models and unite domain experts, IT experts and managers. The growing interest in process mining is illustrated by the Process Mining Manifesto [6] recently released by the IEEE Task Force on Process Mining.

This manifesto is supported by 53 organizations and 77 process mining experts contributed to it. The process mining spectrum is quite broad and includes techniques for process discovery, conformance checking, model repair, and role discovery, bottleneck analysis, predicting the remaining flow time, and recommending next steps. Over the last decade hundreds of process mining techniques have been proposed. A process discovery technique uses as input an event log consisting of a collection of traces (i.e., sequences of events) and constructs a process model (Petri net, BPMN model or similar) that \adequately" describes the observed behavior.

A conformance checking technique uses as input an event log and a process model, and subsequently diagnoses differences between the observed behavior (i.e., traces in the event log) and the modeled behavior (i.e., possible runs of the model). Different process model notations can be used, e.g., BPMN models, BPEL specifications, UML activity diagrams, Statecharts, C-nets, or heuristic nets. MXML or XES (www.xes-standard.org) are two typical formats for storing event logs ready for process mining [2][16]. The incredible growth of event data poses new challenges [17]. As event logs grow, process mining techniques need to become more efficient and highly scalable. Dozens of process discovery [1, 11, 12, 16, 30, 18, 24, 25, 28, 31, 41,54, 60, 61] and conformance checking [6, 13, 14, 15, 22, 29, 31, 42, 43, 51, 59]approaches have been proposed in literature. Despite the growing maturity of these approaches, the quality and efficiency of existing techniques leave much to be desired. State-of-the-art techniques still have problems dealing with large and/or complex event logs and process models.

Whereas traditional process mining techniques focus on the online analysis of solitary processes in steady-state, this paper focuses on multiple inter-related processes that may change over time. [4]Processes may change due to seasonal in sequences, working patterns, new laws, weather, and economic development. Moreover, there may be multiple variants of the same process or the process is composed of sub processes. Existing techniques also cannot handle multiple process variants and/or heterogeneous collections of cases. [6] However, in reality the same process may be used to handle very different cases, e.g., in a care process there may be characteristic groups of patients that need to be distinguished from one another. Moreover, there may be different variants of the same process, e.g., different hospitals execute similar care processes, and it is interesting to compare them. Obviously, it is very challenging to discover and compare processes for different hospitals and patient groups. Unfortunately, traditional techniques tend to focus on a single well-defined process. Cases can be clustered in groups and process models can be compared, however, there are no process discovery techniques that produce overarching models able to relate and analyze different groups and process variants.

III. SPLITTING AND MERGING OF PROCESS CELLS

In this paper, we propose the new notion of process cubes where events and process models are organized using different dimensions (e.g., case types, event classes, and time windows). A process cube may have any number of dimensions used to distribute process models and event logs over multiple cells. The first process cube shown in Figure 3(top) has three dimensions: case type, event class and time window. [8]In this simple example, there is only one case type and only one event class. The cube covers multiple time windows, but only one is shown. (All cases completed in 2012). In this toy example there are only eight cases (i.e., process instances) and seven distinct activities. The process may be split by identifying multiple case types and/or multiple event classes. The second process cube shown in the bottom side of Figure 3 has two case types (gold customer and silver customer) and two event classes (sales and delivery). The case type dimension is based on properties of the case the event belongs to. In Figure 3 (bottom), cases 1, 4, 5 and 6 refer to a \gold customer". Hence, the cells in the gold customer" row include events related to these four cases. The event class dimension is based on properties of individual events, e.g., the event's activity name, its associated resource, or the geographic location associated with the event. In Figure 2 (right), the event class dimension is based on the activity of each event. The event class \sales" includes activities A, B, and C. The event class \delivery" refers to activities C, D, E, F , and G. The time window dimension uses the timestamps found in the event log. A time window may refer to a particular day, week, month, or any other period.

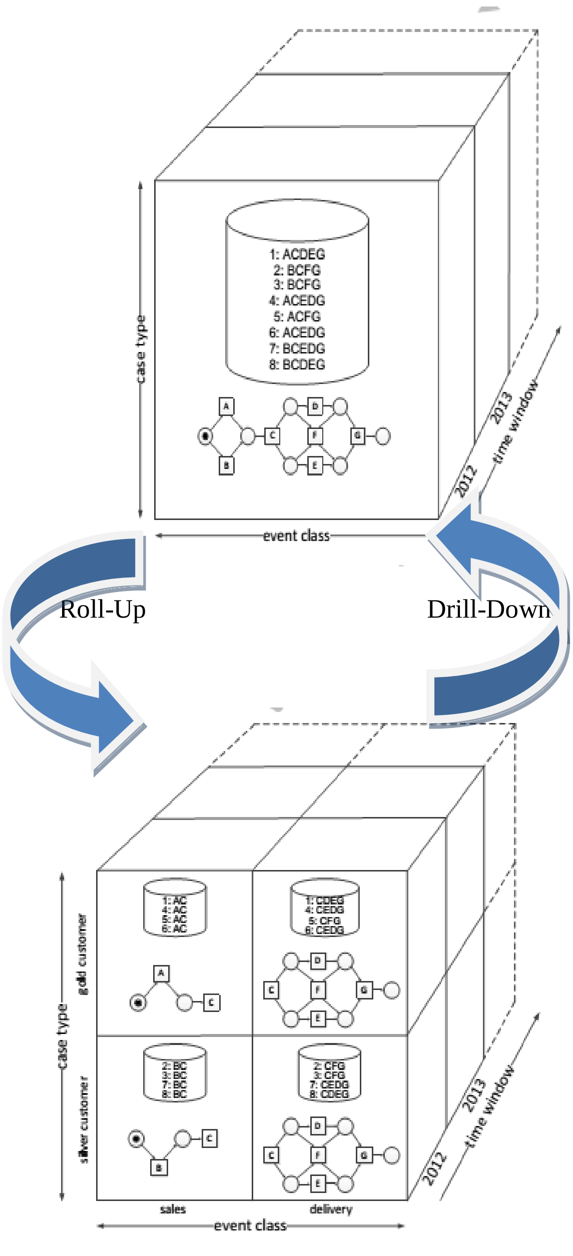


Fig 3) Two process cubes illustrating the splitting (drilling down) and merging (rolling up) of process cells using the case type and event class dimensions.

Each cell in a process cube refers to a collection of events and possibly also processes mining results (e.g., a discovered process model) or other artifacts (e.g., a set of business rules). Events may be included in multiple cells, e.g., sales and delivery cells share C events. Each of the three dimensions may have an associated hierarchy, e.g., years composed of months and months composed of days. Process cubes are related to the well-known OLAP (Online Analytical Processing) cubes [9] and large process model repositories [4]. In an OLAP cube, one can drill-down or roll-up data, zoom into a selected slice of the overall data, or reorder the dimensions. However, OLAP cubes cannot be used for process related data since events are ordered and belong to cases. Moreover, cells are associated to process models and not just event data. Conversely, process model repositories do not store event data. In process cubes, models and event data are directly related. Observed and modeled behavior can be compared, models can be discovered from event data, and event data can be used the breathe life into otherwise static process models. This paper defines OLAP notions such as \slicing", \dicing", \rolling up" and \drilling down" for event data. These can be used to compare, merge, and split process cells at both the log and model level. The process cube notion is closely related to divide-and-conquer approaches in process mining where huge event logs are partitioned into smaller sub logs to improve performance and scalability.

In principle, process cubes can also be used to decompose challenging process mining problems into smaller problems using the techniques described in [3, 5, and 4]. These techniques may be used to speed-up OLAP operations.

IV. PROCESS CUBES

A Process Cube combines the understanding of markets with powerful analytics, to identify the optimal pricing architecture for business purpose, focusing discounts on the most price-sensitive products and customers, where it will have the biggest payoff in competitive positioning; and extracting small premiums on less-sensitive products and customers.

For Example: The Process Cube takes the guesswork out optimized pricing in complex environments, allowing sales to focus on serving the customer and driving value. It provides flexibility and discipline in pricing, to create a balance that works in business issues. Process Cube is illustrated more clearly with an this example ie, stated below :(Refer Fig 4)

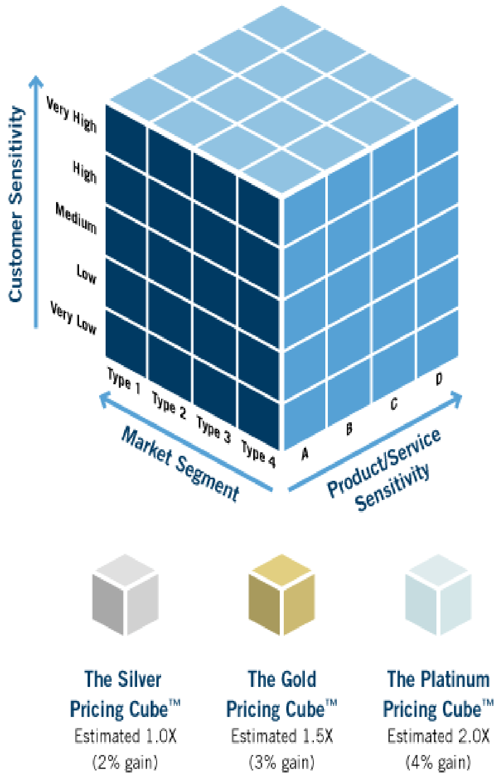


Fig 4: Strategic Pricing of a 3D-Cube.

Every customer is assigned to a market_segment and customer size and/or sensitivity level; every product/service is assigned to a product/service family every product/service is profiled for price sensitivity. Process Cube provides a foundation for setting pricing standards in complex environments. [10.] Process Cube evolves from The Silver Pricing Cube to Gold and Platinum levels, to achieve increasing sophistication and impact.

Silver Pricing Cube builds basic pricing structures based on customer_segment and size; and overall product/service sensitivity ratings Gold Pricing Cube incorporates dynamic premiums; and segment-specific product/service sensitivity ratings. Platinum Pricing Cube profiles advanced customer-sensitivity characteristics; incorporates customer behavioral scoring methods; as well as customer cost-to-serve process. As illustrated by Figure 5, event data can be used to construct a process cube. Each cell in the process cube corresponds to a set of events selected based on the corresponding dimension values. In Figure 4 there are three dimensions. However, a process cube can have any number of dimensions n € N.

Moreover, dimensions cn be based on any event property. In Figure 5 events are grouped in cells based on case type, a particular event class, and a particular time window, \i.e, one cell refers to the set of all events belonging to case type ct, event class ec, and time window tw.

The case type dimension is based on properties of the case as a whole and not on the characteristics of individual events. Hence, if event e is of type ct , then all events of the case to which e belongs, also have type ct . [11]Case type ct may be based on the type of customer (gold of silver) or on the total amount (e.g., < 1000 or >= 1000). The event class dimension is based on properties of the individual events, e.g., the event's activity name, associated resources, or geographic location. Event type (et) may depend on the activity name, e.g., there could be three event classes based on overlapping sets of activity names: {A, B},{C,D}, and {E}.

The time window dimension uses the timestamps found in the event log. A time window (tw) may refer to a particular day, week, month, or any other period, e.g., to all events that took place in December 2012. An event may belong to multiple process cells because case types, event classes, and time windows may be overlapping. Process cells may be merged into larger cells, i.e., event data can be grouped at different levels of granularity. Semantically, the merging of cells corresponds to the merging of the corresponding event sets. One may refine or coarsen a dimension. A process cube is composed of a set of process cells as shown in Fig 5. Per cell one may have a predefined or discovered process model. The process model may have been discovered from the cell's event data or given upfront. Moreover, other artifacts, e.g., organizational models [5], may be associated to individual cells. Process cubes will be used to relate different processes, e.g., we may be interested in understanding the differences between gold and silver customers, large orders and small orders, December and January, John and Ann, etc.

Moreover, we may want to chop a larger cell into many smaller cells for efficiency reasons (e.g., distributing a time-consuming discovery task). The three dimensions shown in Figure 4 only serve as examples and may be refined further, e.g., there may be multiple dimensions based on various classifications of cases (e.g., customer type, region, size, etc.). [12]Moreover, each dimension may have a natural hierarchical structure (e.g., a year is composed of months and a country is composed of regions) that can be exploited for the aggregation, refinement, and selection of event data. Process cells (and the associated sub logs nd models) can be split and merged in two ways as is illustrated in Figure 6. The horizontal dimension of a cell refers to model elements (typically activities) rather than cases. The vertical dimension of a cell refers to cases rather than model elements.

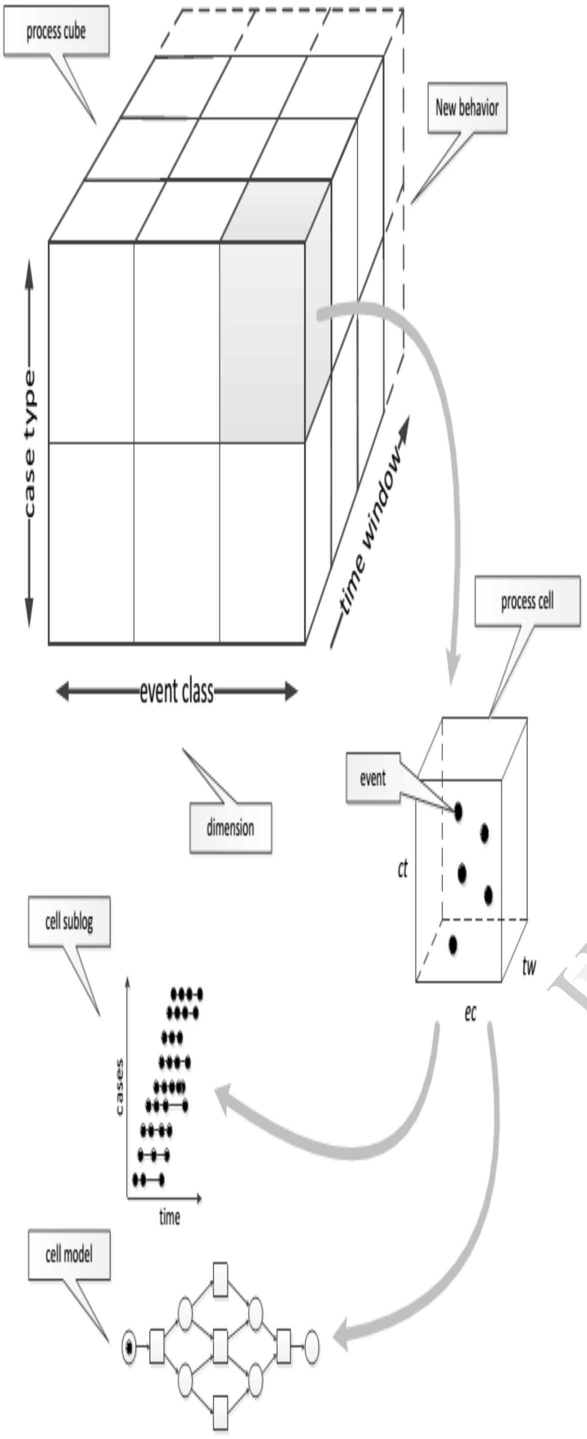


Fig 5: A process cube relates events to different dimensions. Each cell in the cube corresponds to a sub log containing events and may have an associated set of models or other artifacts (derived or given as input).

Figure 5, taken from [4], shows relevant process cube characteristics and is therefore, representative for the definitions of different process cube concepts given below (e.g., process cube, process cell). A detailed discussion on the elements of the Figure 5 is presented in [6]. [11]A process cube is a multidimensional structure built from event log data in a way that facilitates further meaningful process mining analysis. A process cube is composed of a set of process cells [4] and the main difference between a process cube and an OLAP cube lies in its cell characteristics. In contrast to the OLAP cube, there is no real measure of interest quantifying a business operation. While OLAP structures are designed for business operations analysis, the process cube aims at analyzing processes. Therefore, each dimension of analysis is composed of event attributes. Consequently, the content of a cell in the process cube changes from real numbers to events. While in OLAP, dimensions of analysis are used to populate the cube, in case of process cubes the events of an event log are used to create the dimensions of analysis. Note that to differentiate between two events with the same attributes, the event id is added as a dimension of analysis. Consequently, for each event there will be a unique combination of dimension of analysis members.

A process cell can be defined as a sub cube obtained by slicing each of the process cube dimensions. Let PC, PC = (CS, CA). The process cell is slice$_{1,v1}$ (slice$_{2,v2}$ … (Slice$_{n-1}$,v$_{n-1}$(slice $_{n-1,}$v$_{n}$(P C)))…)) = P C^{1}**.**Each cell in the process cube corresponds to a set of events [4], returned by the cell event function CE. The process cube, as defined above, is a structure that does not allow overlapping of events in its cells. To allow the comparison of different processes using the process cube, a table of visualization is created. The table of visualization is used to visualize [14] only two dimensions at a time. Multiple slice and dice operations can be performed by selecting different elements of the two dimensions. Each slice, dice, roll-up or drill-down is considered to be altering operation. Hence, a new filter is created with each OLAP operation. Filters are added as rows/columns in the table of visualization. Note that unlike the cells of the process cube, the cells of the table of visualization may contain overlapping events. That is because there is no restriction in selecting the same dimension members for two filtering operations. Given a process cube *PC* , a process model, M$_{PC}$ is the result of a process discovery algorithm, such as Alpha Miner, Heuristic Miner or other related algorithms, used on *PC* . However, [15] there are various process mining algorithms whose results are not necessarily process models. Instead, they can offer some insightful process-related information. For example, Dotted Chart Analysis provides metrics (e.g., average interval between events) related to events and their distribution over time. Process cubes are not limited to process models as well. Therefore, we refer to process mining results just as models.

So far, we described the process cube as being a hypercube structure, with a finite number of dimensions. In [4], a special process cube is presented, with three dimensions: case type (ct), event class (ec) and time window (tw).

V. EVENT LOG CITATION

Figure 6, taken from [4], contains a table corresponding to a fragment of an event log. Let the event data from the event log be used to construct a process cube *PC*. [16]Then, the *ct*, *ec* and *tw* dimensions are established as follows. The case type dimension is based on the properties of a case. For example, the case type dimension can be represented by the type of the customer, in which case, the members of ct are gold and silver, i.e. D1 = {gold; silver} H1 = D1. The event class dimension is based on the properties of an event. For example, ec can be represented by the resource and include, as such, the following members: D2 = {John}, H2 = D2.The time window dimension is based on timestamps. A time window can refer to years, months, days of week, quarters or any other relevant period of time. Due to its natural hierarchical structure, *tw* dimension can be organized as a hierarchy, e.g., 2012 → 2012Dec →2012DecSun. We consider D3 = {2012DecSun} and H3 = {2012, 2012Dec, 2012DecSun}.

Let D1 = {gold; silver}, D2 = {John} and D3 = {2012DecSun}

H1 = {gold; silver}, H2 = {John} and H3 = {2012, 2012Dec, 2012DecSun}

CD = D1× D2×D3 be the cube dimensions,

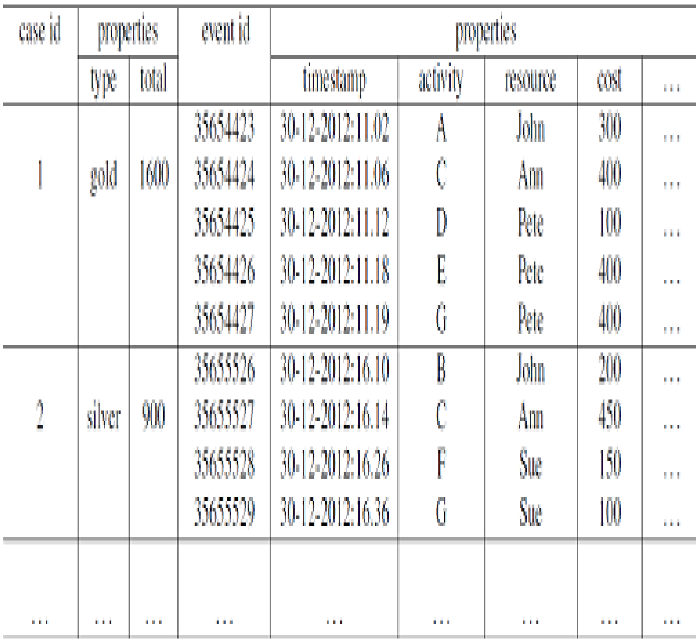
CH = H1×H2×H3 be the cube hierarchies,

h_1, h_2 € H_3, h_1= 2012, children (h_1) = {2012Dec}, h2= 2012Dec, children (h_2) =2012DecSun.

h_1,h_2€ H_3,h1= 2012,all Leaves(h_1) = {2012DecSun}, h2= 2012Dec, all Leaves(h_2)= 2012DecSun,

CS = (CD, CH) be the process cube structure,

h_1€H_1,h_1=gold, all Leaves (h1) = {gold}, h_2€H_2,h_2 = John, all Leaves (h_2) = {John}, h_3€H_3,h_3= 2012,all Leaves (h3) = {2012DecSun},CE (h1, h2, h3) = {35654423},CC (35654423) = (gold, John, 2012DecSun). For the rest of the elements of CH, CE is defined in the same way.



case id	properties		event id	properties				
	type	total		timestamp	activity	resource	cost	...
1	gold	1600	35654423	30-12-2012:11.02	A	John	300	...
			35654424	30-12-2012:11.06	C	Ann	400	...
			35654425	30-12-2012:11.12	D	Pete	100	...
			35654426	30-12-2012:11.18	E	Pete	400	...
			35654427	30-12-2012:11.19	G	Pete	400	...
2	silver	900	35655526	30-12-2012:16.10	B	John	200	...
			35655527	30-12-2012:16.14	C	Ann	450	...
			35655528	30-12-2012:16.26	F	Sue	150	...
			35655529	30-12-2012:16.36	G	Sue	100	...
...	...	...	...	...	...	...	...	...

Fig 6: Event Log selection

VI. CONCLUSION

In this paper, we pompous the notion of process cubes. It gives end users the opportunity to analyze and explore processes interactively on the basis of a multidimensional view on event data. No need to extract event logs a head of time like in traditional process mining approaches.

REFERENCES

[1] W.M.P. van der Aalst. Process Mining: Discovery, Conformance and Enhancement of Business Processes. Springer-Verlag, Berlin, 2011.

[2]. W.M.P. van der Aalst. Decomposing Process Mining Problems Using Passages. In S. Haddad and L. Pomello, editors, Applications and Theory of Petri Nets 2012, volume 7347 of Lecture Notes in Computer Science, pages 72{91. Springer-Verlag, Berlin, 2012.

[3]. W.M.P. van der Aalst. Distributed Process Discovery and Conformance Checking. In J. de Lara and A. Zisman, editors, International Conference on Fundamental Approaches to Software Engineering (FASE 2012), volume 7212 of Lecture Notes in Computer Science, pages 1{25. Springer-Verlag, Berlin, 2012.

[4] W. M. P. van der Aalst. Mining Process Cubes from Event Data (PROCUBE), project proposal (under review). 2012.

[5] W. M. P. van der Aalst. Process Mining: Making Knowledge Discovery Process Centric.SIGKDD Explorations Newsletter, 13(2):45{49, 2012.

[6] W. M. P. van der Aalst. Process Cubes: Slicing, Dicing, Rolling Up and Drilling Down Event Data for Process Mining. In J. Liu M. Song, M.Wynn, editor, Asia Pacific conference on Business Process Management (AP-BPM 2013), Lecture Notes in Business Information Processing, 2013.

[7] C.W. Gunther and W.M.P. van der Aalst. Fuzzy Mining: Adaptive Process Simplification Based on Multi-perspective Metrics. In G. Alonso, P. Dadam, and M. Rosemann, editors, International Conference on Business Process Management (BPM 2007), volume 4714 of Lecture Notes in Computer Science, pages 328-343.Springer-Verlag, Berlin, 2007.

[8]. M. Hilbert and P. Lopez. The World's Technological Capacity to Store, Communicate, and Compute Information. Science, 332(6025):60-65, 2011.

[9]. IEEE Task Force on Process Mining. Process Mining Manifesto. In F. Daniel, K. Barkaoui, and S. Dustdar, editors, Business Process Management Workshops, volume 99 of Lecture Notes in Business Information Processing, pages 169-194.Springer-Verlag, Berlin, 2012.

[10]. M. van Leeuwen and A. Siebes. StreamKrimp: Detecting Change in Data Streams. In Machine Learning and Knowledge Discovery in Databases, volume 5211 of Lecture Notes in Computer Science, pages 672{687. Springer-Verlag, Berlin, 2008.

[11]. C. Li, M. Reichert, and A. Wombacher. The MINADEPT Clustering Approach for Discovering Reference Process Models Out of Process Variants. International Journal of Cooperative Information Systems, 19(3-4):159-203, 2010.

[12]. T. Mamaliga. Realizing a Process Cube Allowing for the Comparison of Event Data. Master's thesis, Eindhoven University of Technology, Eindhoven, 2013.

[13]. J. Manyika, M. Chui, B. Brown, J. Bughin, R. Dobbs, C. Roxburgh, and A. Byers. Big Data: The Next Frontier for Innovation, Competition, and Productivity. McKinsey Global Institute, 2011.

[14]. A.K. Alves de Medeiros, A.J.M.M. Weijters, and W.M.P. van der Aalst. Genetic Process Mining: An Experimental Evaluation. Data Mining and Knowledge Discovery, 14(2):245{304, 2007.

[15]. J. Munoz-Gama and J. Carmona. A Fresh Look at Precision in Process Conformance. In R. Hull, J. Mendling, and S. Tai, editors, Business Process Management (BPM 2010), volume 6336 of Lecture Notes in Computer Science, pages 211-226. Springer-Verlag, Berlin, 2010.

[16]. J. Munoz-Gama and J. Carmona. Enhancing Precision in Process Conformance: Stability, Confidence and Severity. In N. Chawla, I. King, and A. Sperduti, editors,IEEE Symposium on Computational Intelligence and Data Mining (CIDM 2011), pages 184{191, Paris, France, April 2011. IEEE.

[17] A. Sheth. A New Landscape for Distributed and Parallel Data Management. Distributed and Parallel Databases, 30(2):101-103, 2012.

AUTHORS BIBLOGRAPHY:

Sarada Sowjanya.C completed her M.Tech from JNTUK with Distinction. She did her BE from RASTRASANT TUKADOGI MAHARAJ NAGPUR UNIVERSITY, NAGPUR Dt (Maharashtra).Her areas of interest are: Data Mining, Computer Networks, Data Base Management Systems.

J.Sowjanya completed her M.Tech from JNTUH with Distinction. She did her BE from OU.Her areas of interest are: Data Mining, Web Technologies.

T.Charan Singh completed his MTech from JNTU. He did his BTech from JNTU.His areas of Interest are Data mining, OOAD.

A.Rangamma has completed her MTech from IETE. She completed her MCA from Andhra University. Her areas of interest are Data Mining, Computer Networks and Data Structures.