Everything You Always Wanted to Know About Your Process, But Did Not Know How To Ask Eduardo González López de Murillas Hajo Reijers Wil van der Aalst

International Workshop on Process Querying 2016 September 19th, 2016 Rio de Janeiro, Brazil

> Technische Universiteit **Eindhoven** University of Technology

EXPERIAL.

Where innovation starts

TU





Making sense out of the big mess



Can we actually "know" the universe? My God, it's hard enough finding your way around in Chinatown.



We need to find the needle in the haystack





Look in the right place



Querying can help us





Simple example



Customers for which the status value has changed from Silver to Gold membership, and are linked to an office located in Madrid.



Using SQL on the underlying Meta Model

SELECT OBJCust.id FROM object as OBJCust, class as CCust. class as COff. version as OBJVSilver, version as OBJVGold. version as OBJVMadrid, attribute value as ATVStatusSilver, attribute value as ATVStatusGold, attribute name as ATStatus, attribute value as ATVLocation, attribute name as ATLocation, object as OBJOff, class as COff. version as OBJVOff, attribute value as ATVLocation, attribute name as ATLocation, relation as REL WHERE OBJCust.class id == CCust.id AND CCust.id == ATStatus.class id AND ATStatus.name == "status" AND ATVStatusGold.object version id == OBJVGold.id AND ATVStatusSilver.object version id == OBJVSilver.id AND ATStatus.id == ATVStatusGold.attribute name id AND ATStatus.id == ATVStatusSilver.attribute name id

- AND ATVStatusGold.value == "Gold"
- AND ATVStatusSilver.value == "Silver"
- AND OBJVGold.object_id == OBJCust.id
- AND OBJVSilver.object_id == OBJCust.id
- AND OBJVGold.start_timestamp >= OBJVSilver.end_timestamp
- AND COff.name == "OFFICE"
- AND OBJOff.class_id == COff.id
- AND OBJVOff.object_id == OBJOff.id
- AND ATLocation.name == "Location"
- AND ATLocation.class_id == COff.id
- AND ATVLocation.attribute_name_id == ATLocation.id
- AND ATVLocation.object_version_id == OBJVOff.id

AND (REL.source_object_version_id == OBJVOff OR REL.target_object_version_id == OBJVOff) AND (REL.source_object_version_id == OBJVSilver OR REL.source_object_version_id == OBJVGold) AND (REL.target_object_version_id == OBJVSilver OR REL.target_object_version_id == OBJVGold)



In DAPOQ-Lang

objectsOf(versionsRelatedTo(versionsOf(allClasses where name == "OFFICE") where at.Location == "Madrid") where at.status changed from "Silver" to "Gold"

> Customers for which the status value has changed from Silver to Gold membership, and are linked to an office located in Madrid.



Main Trends on Querying Execution Data

- Data Provenance Oriented
 - Origin or source of data



- Business Process Oriented
 - Event data
 - Process models



			Inț	put d	ata	_	P	rove	nanc	ce	Bb	P DB				Query aspects								
Ref	Title	Event data	Model-based	Storage model	Complex event processing	Streams	Provenance-oriented	OPM-compliant	Data lineage	Depencency graphs/Relations	Business process oriented	Entities/Artifacts	Database-oriented	State-aware	Graph-based	Relevance querying	Semantic querying	Regular path queries (RPQ)	Projection queries	femporal properties/querying	Event correlation	Multi process	Multi log	
[3]	Answering regular path queries on workflow provenance	\checkmark					\checkmark			-					\checkmark			\checkmark					_	ſ
[4]	Capturing and querying workflow runtime provenance with PROV: A practical approach	~		~			~			~														ſ
[5]	Modeling and querving scientific workflow provenance in the D-OPM	1		5			5	1			-	\vdash			1		\vdash	5					-	H
[6]	Towards a scalable semantic provenance management system	1					./	1	\checkmark	\checkmark		\vdash				-	1		\vdash	\vdash			-	H
[7]	Towards integrating workflow and database provenance	\checkmark		\checkmark			~	\checkmark	\checkmark	<i>√</i>		\checkmark	\checkmark	\checkmark		-			\vdash	\vdash			-	t
[8]	MTCProv: A practical provenance query framework for many-task scientific computing	√		√			1	 ✓ 	√ -	~														ſ
[9]	OPQL: A first OPM-level query language for scientific workflow provenance	√					~	~	~						~	Ĺ		F						ſ
[10]	Storing, reasoning, and querying OPM-compliant scientific workflow	~		~			~	~	~	✓														ſ
[11]	XOuerv meets Datalog: Data relevance query for workflow trustworthiness	5			-		5			1	-				-	1		-					-	t
[12]	A model for user-oriented data provenance in pipelined scientific workflows	√		\checkmark			· √		\checkmark	· √								-					-	t
[13]	A knowledge driven approach towards the validation of externally acquired traceability datasets in supply chain business processes	√							· •		~					\square	V							ſ
[14]	Process query language: A way to make workflow processes more flexible	\checkmark		\checkmark					\checkmark		\checkmark					\vdash			\vdash				-	t
[15]	Workflow History Management	√		√												+	-	-					-	t
[16]	The HIT model: Workflow-aware event stream monitoring	√	\checkmark		\checkmark	\checkmark					· √					\vdash	\vdash	-					-	t
[17]	Semantic Enabled complex Event Language for business process monitor- ing	· •			V				V		· •				√	Ĺ	V		√	~				ľ
[18]	A framework supporting the analysis of process logs stored in either relational or NoSQL DBMSS	~		V							~				V									ſ
[19]	Business impact analysis: a framework for a comprehensive analysis and optimization of business processes	√		~						~	~	~	\checkmark											ſ
[20]	Model-driven event query generation for business process monitoring	\checkmark	\checkmark		\checkmark						\checkmark												-	t
[21]	Querying process models based on the temporal relations between tasks		\checkmark								\checkmark						\square			\checkmark			_	ſ
[22]	A query language for analyzing business processes execution	\checkmark							\checkmark	\checkmark	\checkmark	\checkmark			\checkmark	Ē	\checkmark	í –			\checkmark		-	Γ
[23]	Top-k projection queries for probabilistic business processes	\checkmark	\checkmark	\checkmark							\checkmark								\checkmark					Γ
[24]	Integration of Event Data from Heterogeneous Systems to Support Business Process Analysis	~	V	~							~										V			[
[25]	Enabling semantic complex event processing in the domain of logistics	\checkmark			\checkmark						\checkmark						\checkmark			\square			_	ľ
[26]	Optimizing complex sequence pattern extraction using caching	\checkmark			\checkmark						\checkmark								\checkmark				_	ľ
[27]	Log-based understanding of business processes through temporal logic query checking	~																		✓				ſ
	Our approach																		\square					ĺ





- **1.** Combination of data and process aspects
 - **2.** Support for database particularities
 - **3.** Integration in the analysis workflow



Why do we need another query language?

SQL is to Process Querying what Assembly is to programming.



DAPOQ-Lang

- Data Aware Process Oriented Query Language
 - Considers data and process perspectives
 - State aware
 - Easier to make common queries
 - Possible to combine with other analysis or filtering techniques (Integrated in the analysis workflow)



DAPOQ-Lang I



DAPOQ-Lang II

Functions	
datamodelsOf	
classesOf	
attributesOf	
relationshipsOf	
objectsOf	
versionsOf	Returs the set of requested elements (datamodels,
relationsOf	classes, attributes, etc) related to the input elements.
eventsOf	Input can be any of the primitive elements (and periods)
casesOf	
logsOf	
activityInstancesOf	
activitiesOf	
processesOf	I w I hiversity of Technolog

DAPOQ-Lang III

Functions	
versionsRelatedTo	Versions related to versions
periodsOf	Periods have a start and end timestamp
concurrentWith	With scopes: same case, same process, same object

Set Operations	
INTERSECTION	
UNION	On sets of elements of the same type: cases with cases versions with versions etc
EXCLUDING	

Other constructs	
Foreach	On elements of a set
If Then Else	Condition checks if set is empty
Filters	On attributes of elements (versions, events, cases)
	University of rechnology

DAPOQ-Lang IV

Filters	
Common attributes	WHERE name == "CUSTOMER"
Specific attributes	WHERE at.Customer_name CONTAINS "Edu"
Versioning of values	WHERE at.Address CHANGED FROM "Madrid" to "Barcelona"
Scopes (Only for concurrency by now)	Select elements of same nature that share a common link to another type: concurrentWith(allActivities where name == "Pay")
	scope case



Can we ask anything?



I don't know the question, but sex is definitely the answer.



We can express simple questions

allVersions() where { at.PRICE == "35" }



PQ'16 - 2016-09-19

Or complex ones

```
def allC = allCases().where { id < 10} ;</pre>
```

```
def results = allActivities().where { false };
```

```
def eventsN = eventsOf(allActivities().where{ name.contains("Afwijzing") });
```

```
for (Object c : allC.result) {
    def ec = eventsOf(c);
    def e = eventsN.intersection(ec);
    def pe = globalPeriod(e);
    def ebef = ec.where { before(globalPeriod(it),pe) };
    results = results.union(activitiesOf(ebef));
}
```

return results



We do not need to have all the answers

• If we get closer to the truth



We do not need to have all the answers

- If we get closer to the truth
- DAPOQ-Lang makes simple to select the relevant portion of the data



We do not need to have all the answers

- If we get closer to the truth
- DAPOQ-Lang makes simple to select the relevant portion of the data











😕 🗏 🔲 //Local I	Repository/m	etamodel-	creation/b	lock – Ra	apidMiner	Studio Ba	asic 7.2	ı @ 000.	nbwin	1268										
<u>F</u> ile <u>E</u> dit <u>P</u> rocess	<u>V</u> iew <u>C</u> onne	ctions <u>S</u> ett	ings E <u>x</u> ter	isions																
		\subset	•				Views:	D	esign	Res	ults								🚱 Qu	estions? 💌
Result History	×	SLEXMMIC	00bject (S		Meta Moc	lel Genera	ator (2)) ×												
sposi	org.processmining.openslex.metamodel 🔻					Name Pro						Refresh								
- 1	Activity Id Na		Name			1 log01	log01		Id	Name		Activit	Activ	it	Event Id	d Orderi		Attrib	Value	Туре
Meta medel vieuelizer		1 Call	OUT	~						1 10008.0	~	1		1 ^		1	1 ^	CustomerID 1	0009.0	3
model visualizer		2 Nog	geen beslis	ssing						210009.0		2	2	1		2	1	Event C	all OUT	6
		3 Toe	kenning	=						310022.0		3	•	1		3	1	EventDat 2	016-04	. 7
		4 Afwi	jzing							4 10026.0	_	4		1		4	1	Ind_CallO 0	.0	3
		5 Init						I		510042.0	_	5		1		5	1	TypeOfCall T	eletoon	7
		6 Upg 7 Coo	aveontvang	en						610090.0	_	6		1	1	6	11*	Event Life n	ull	STRING
		7 Gec	ontroleeru	eina				L		/1009/.0					AI	l Events		Event Re n	450046	LONG
		9 Toe	kenning	sing				L		810106.0	_		8				-1	Event min 1	439040	LUNG
		10 Afwi	izina							910113.0		All Activi	All Activity Instances			Per Activity				
		11 Lett	er OUT						1	1 10121.0		Por	Activity		Por Ac	tivity Instance	,			
		12 WN	ww		-	All Logs				210120.0	\vee		ACCIVICY		FEIAU	livity instance	-			
		13 WN	WW Afhande	eling	Logs	Per Proces	s	All Ca	ses	Per Log		Pe	r Case		P	er Case				
		14 UW	7 Algemeen		L										P					
	Maximize					Obje	ct (Class Id		Version	Obje	ct Id Star	t Ti	End Tin	n Att	Attribute N		alue	Туре	
			30297			з	62125		30302 14413	24955	1447670	183 IKO T	imeStamn	201	6-07-01 14-2	7				
						3	80298		3	62126		30302 1447	30302 144724955 14470708			ate Receive	1 201	15-11-16 09:3 7		
							30299		3	62127		30302 1447	75338	1447753	312 IKO S	tatus	Geo	econtroleerd 7		
	The Physical Jack					3	30300		3	62128		30302 1447	75312	1467381	79 IKO S	ubStatus	TeV	/errekenen	7	
	121 Billionaria	_					30301 30302		3	62129		30302 14673	30302 146738179 1467382		210 IKO_S	endin_From	. 201	015-10-01 6		
	121 Beldelberkan 131 Beldelaum								3	62130		30302 14673	30302 146738210 1467378			0 IKO_PeriodEnd 20		015-09-30 6		
	191 Index Statement of a				0303		3	62131		30302 146737810			-1 IKO_P	eriodStart	201	.5-09-01	6			
	1.01 Balanses				30305		3						IKO_C	hangeType	U		6			
	1.01 Automation Cars 1.01 Service and advantage of the Cars 1.01 Service and the Cars						3						IKO_I)	127	7002.0	3			
	181 billade politik 181 billade Joseff (Bella	M					30307		3						Custo	omeriD	610	159.0	3	
	141 Manual Johnson						80308		3											
	Ini titaphorisasan	and agreent				3	30309		3											
		24-13		MUND - Millered		3	80310		3	Relation Id		Relationshi	. s	ource Obi	ie Tar	aet Obje	St	tart Timest	End T	imesta
	Market 1	20 Octow/D (40 36,0	Service, Self 14	di falsa,3 di falsada			30311		3	70	022		1			0004	0 1	44775010600	0 1467	201700265
	It belowed IT BOMont.3 It belowing It Boladore It had It consider	14 Myniatur 14 Myniatur	(8) bried (8) bellicitie (8) bel	M rgathaus M rgathaus	(m) harmenta		80312		3	72	932		1	e e	32120	9084	9 1 9 1	46738179826	5 1467	382105783
	St Sadial Inc. Inc. Inc. Inc. Inc. Inc. Inc. Inc.	14: Michaels, Practan 14: Michaelses 06: Michaelses	UR MANY	Al Antonials Al Antonials Al Antonials	Hi hat		0313		3	72	934		1	6	52130	9884	91	46738210578	3] 467	378102457
	Lances	D Hotelese D Hotelese		OI NALIATING			0314		3	72	935		1	6	52131	9884	9 1	46737810245	7	-1
	R.M. W. Laborer	0 (NAMA AND		II NOTICE AND I	/ norman a		80316		3	72	936		1		52125	9884	9 1	44124955381	9 1447	670830793
and a contract of the contract			30317		3	72	937		1	e	52126	9884	91	44767083079	3 1447	753386075				
							80318		3	72	938		1	e	52127	9884	91	44775338607	5 1447	753126308
							0319		$_{2}$ $^{\vee}$											
							All Ob	jects												
							Per C	lass												
						?														





😣 🖻 🗊 //Local Repository/metamodel-creation/query-test-2 – RapidMiner Studio Basic 7.2.000 @ nbwin1268												
<u>F</u> ile <u>E</u> dit <u>P</u> rocess <u>V</u> iew <u>C</u> onnections	Settings Extensions											
	Views: Design Results	Questions? 💌										
Repository X	Process X	Parameters X										
C Add Data ≡ ▼	Drocess) 100% @ O O Z a	Process										
▶ 🔁 Samples	Pr	logverbosity init • ①										
Local Repository (eduardo)	gui. dialog.parameter. dapoql-query-editor. title: configure operator Configure this operator by means of a Wizard.	res logfile 🔁 🗈										
 Celonis-challenge-3 (eduardo) celonis-challenge-4 (eduardo) 		res										
🕨 🔁 dapoq-lang (eduardo)	<pre>var _cust = objectsOf(allclasses where name == "CUSTOMER"); </pre>											
data (eduardo) metamodel-creation (eduardo)	<pre>var _ikos = objectsOf(allclasses where name == "IKO");</pre>											
processes (eduardo)	<pre>var _custIkos = objectsOf(versionsrelatedto (versionsOf(_ikos))) intersection _cust ;</pre>											
	return versionsOf(_custIkos)											
Operators X												
inductive X		Show advanced parameters										
Extensions (5)		Change compatibility (6.0.002)										
TapidProM (5)												
Analysis (1)		Help ×										
Inductive Visual Miner Discovery (2)		Process										
Inductive Miner (Petri Net)		RapidMiner Studio Core										
Inductive Miner (Process T T Streams (2)		Synopsis										
 Streams (2) Discovery (2) 	✓ <u>o</u> K X <u>Cancel</u>	The root operator which is the outer most operator of every process.										
Stream Inductive Miner (Ac		Description										
		Each process must contain exactly one										
<		operator of this class, and it must be the root operator of the process. This										





Conclusion

- DAPOQ-Lang provides
 - Simplicity
 - Compactness
 - Suitability
- Can be used together with SQL
- Allows to focus on the analysis
- Makes easier to query and filter information
 - In the scope of Business Process Execution Data
- Integrated in a bigger tool:
 - One more element in the analysis workflow



Future work

- The function collection is not complete
 - Evaluation with users to asses usefulness
 - Incorporate new constructs to simplify analysis
- Improve performance
- Include more operators in RapidProM to interact with other analysis tools



Questions



