Designing Enterprise System Information Architecture Using Task Data

Dawei Huang(✉)

Microsoft Corporation, Redmond, WA, USA
whuang@microsoft.com

Abstract. Much of today's information architecture for enterprise tools is organized in a complex and feature oriented way – frustrating the users and requiring tons of learning to be efficient. We introduced a task taxonomy research project that studied a few hundred database users to understand their responsibilities, roles, and tasks. We created a task taxonomy model filled with large quantitative data on tasks. We found task hierarchy emerged from the model and it had a strong relationship with user roles – implying role-based workspace design principles. With the task hierarchy data, the degree of relevance of task hierarchy could be estimated and applied for enterprise information architecture designs.

Keywords: Task taxonomy · Information architecture · Enterprise · Database management tasks · Database users

1 Introduction

Much of today's information architecture (IA) for enterprise server tools was organized in a complex and feature oriented way. What the users were interacting with was arranged by features or objects, in a manner that was easy to make but hard to use. For years, customer feedback on our database management tools included features hard to find, too many tools with similar functions, tools not integrated, difficult to customize for different users, and unable to scale for managing thousands of servers.

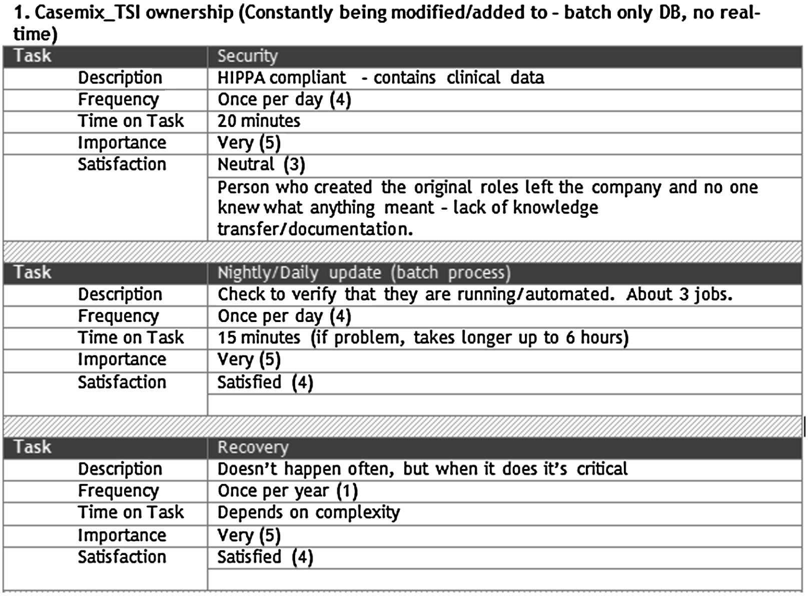
In response, the 'Task Taxonomy' project was started in 2006. It was to present and codify a view of database users' entire task space and to discover user task patterns to inform enterprise IA designs. In the following years, data on user roles, responsibilities and tasks with task frequency, time on task, task difficulty and task importance measures were collected. The task data were analyzed and card-sorted into a rigorously defined Task Taxonomy Model, and loaded into an OLAP data cubes for further exploration.

2 Background: Task Taxonomy Project

After "Task Taxonomy" project was started, standardized data collection protocols and measures were formulated and used. Over 500 database users from 240 companies worldwide were interviewed with data collected on their roles, job responsibilities,

F.F.-H. Nah and C.-H. Tan (Eds.): HCIB 2015, LNCS 9191, pp. 469–480, 2015.
DOI: 10.1007/978-3-319-20895-4_43

tasks and pain points. For each task on which a user reported, measures on frequency with which the task was performed, average time that the task takes, and ratings of importance, complexity, and satisfaction (5-pt or 7-pt Likert scale) were collected during the interview. Example tasks and associated metrics are shown in Fig. 1.



1. Casemix_TSI ownership (Constantly being modified/added to - batch only DB, no real-time)

Task	Security
Description	HIPPA compliant - contains clinical data
Frequency	Once per day (4)
Time on Task	20 minutes
Importance	Very (5)
Satisfaction	Neutral (3)
	Person who created the original roles left the company and no one knew what anything meant - lack of knowledge transfer/documentation.
Task	Nightly/Daily update (batch process)
Description	Check to verify that they are running/automated. About 3 jobs.
Frequency	Once per day (4)
Time on Task	15 minutes (if problem, takes longer up to 6 hours)
Importance	Very (5)
Satisfaction	Satisfied (4)
Task	Recovery
Description	Doesn't happen often, but when it does it's critical
Frequency	Once per year (1)
Time on Task	Depends on complexity
Importance	Very (5)
Satisfaction	Satisfied (4)

Fig. 1. Example of a user reported responsibility and tasks

The task data collected from those 500 users was analyzed in two stages allowing us to derive fuller value. The first stage was to construct a rigorously defined Task Taxonomy model. The second was to use SQL Server's Analysis Services to place the data into OLAP cubes that could then be data-mined. As such, the project has gone through a few activities of taxonomy literature research, data cleansing and card sorting, and discussions with engineering program managers, architects and internal experts to refine the Task Taxonomy model, data cube creation and finally insight reporting.

3 Development of Task Taxonomy Model

A taxonomy is often simply defined as a "systematic classification of information" [1], which operates as experts' classification in industry practices. "Taxonomies can be defined as sets of rules and principles to ensure consistent classification of data and information into ordered categories, attempt to address the problem of information overload. A good taxonomy will bring order and cohesiveness to information portals, thereby speeding up relevant information retrieval and improving business efficiency"

[2]. “Taxonomies are the classification scheme used to categorize a set of information items. They represent an agreed vocabulary of topics arranged around a particular theme. … we typically encounter hierarchical taxonomies such as in libraries, biology, or military organizations” [3]. These definitions match well with the classification research found in library science [4].

In defining Task Taxonomy for this project, we combined those approaches as we relied on both experts’ and users’ knowledge to define the task hierarchy reflected in the database structure. We also set up rules and principles to ensure consistent classification of task information across various input sources.

3.1 Task Taxonomy Model

The database users’ task taxonomy model composes of *task hierarchy, action-object map, and perspectives* (see Fig. 2). A user self-reported task is typically a node in the tree of the *task hierarchy* - a tree of different levels of tasks following a set of hierarchy rules [4]. A low level task can be mapped on to a database UI action-object map. Perspectives such as related technologies, features, job roles, and lifecycle processes are different angles of looking at (or associating with) certain parts of the task hierarchy.

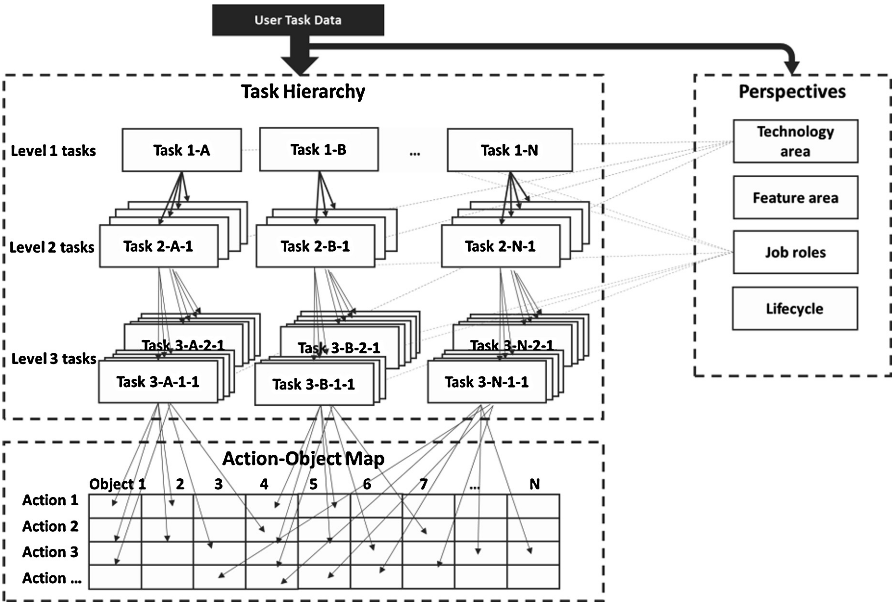


Fig. 2. Task taxonomy model diagram

Task Hierarchy. User’s responsibility and task data were first categorized and mapped to a task hierarchy given a set of strong hierarchy rules [4]:

- *Inclusiveness*: The top class is the most inclusive class and describes the domain of the classification.
- *"Is A" relationship*: A sub-class task is a member of a top-class task.
- *Inheritance*: Attributes of top class tasks are inherited by the sub-class and sub-sub-class tasks.
- *Necessary and sufficient criteria*: To belong to a class, a task must have the necessary and sufficient attributes.

A few other strong hierarchy rules such as mutually exclusiveness were not followed in this model as many user tasks can fall under more than one category as database operational verbs were standardized to fewer common terms.

Action-Object Map. Each of the level 3 tasks in the task hierarchy can be mapped on to *an action-object map* to bridge user tasks to the set of operations entailed. It can be a one-to-one mapping or a one-to-many mapping if the task relates to more than one action or set of objects.

An atomic level of a user task can be expressed as:

$$\textbf{\textit{If}}\,\text{Action(s) X Object(s)} = \text{Operation},$$

$$\textbf{\textit{Then}}\,\text{N (Operation)} = \text{A set or series of operations} = \text{Task}$$

Simplifying a task can involve simplifying actions and object pairings, and/or reducing the number of each by creating macros and combinations of action-object pairings.

The *action-object map* does not imply verb-object order in the sense of a UI. Similar to the way you can use the active or passive voice, e.g., the ball was hit by the boy, the boy hit the ball. The *action-object map* doesn't informs task sequencing either - whether the object is selected first and then the action is applied, or the action is selected first and the object is applied.

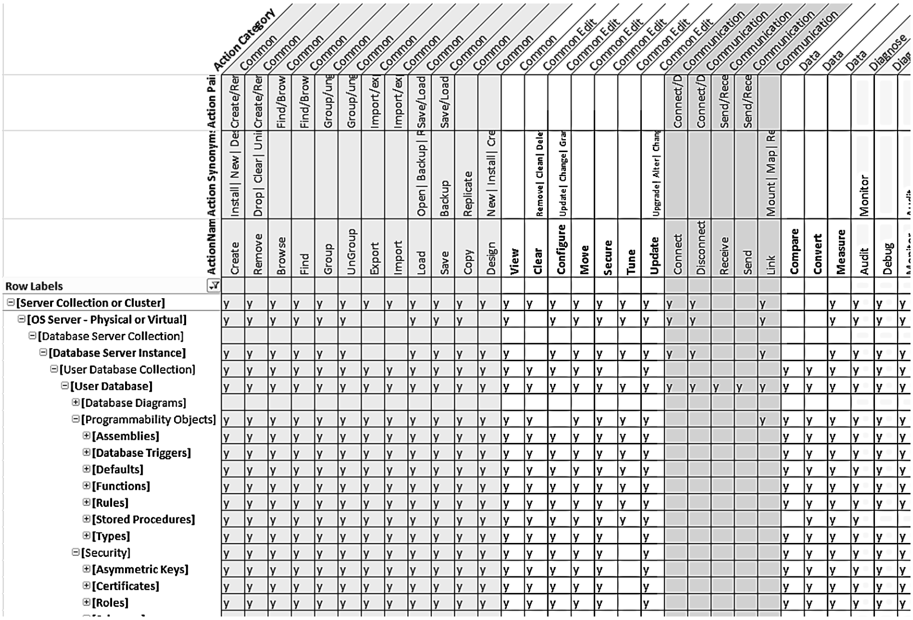
To control the differences in the descriptions that participants used to describe their tasks using natural language, we consolidated synonymous verbs. Verbs with the same or similar meaning were consolidated as action synonyms (see Table 1).

The interaction of actions and objects forms the *action-object map* that can be associated with a level-3 task (see Fig. 3). Given the nature of objects, not all actions are applicable for all objects. For example, it doesn't make sense to apply "connect" or "disconnect" to objects such as stored procedures, types, rules, functions, or triggers. As we observed, the density of an *action-object map* might grow or diminish over time, as users' tasks and technologies evolve.

Perspectives. Perspectives are aspects with which a task is associated, such as *technology area, feature area, job role, lifecycle stages, company size, industry*, and so on. These perspectives are associated with different sets of tasks and objects/actions at different levels, with different densities. For example, database administrator (DBA) is a *job role* perspective whose task space is associated primarily with database management tasks and database server objects. A developer role may be associated with database design, query, debugging, testing related tasks, actions and objects. Similarly,

Table 1. Example of consolidated actions

Action name	Action synonyms	Paired with
Browse	Drill down | Locate	Search/Browse
Copy	Replicate	
Create	Install | New | Design | Code	Create/Remove



Action Category	Common	Common	Common	Common	Common	Common	Common	Common	Common	Common	Common	Common	Common	Common Edit	Common Edit	Common Edit	Common Edit	Common Edit	Common Edit	Communication	Communication	Communication	Communication	Communication	Data	Data	Data	Diagnose	Diag	
Action Pai	Create/Rer	Create/Rer	Find/Brow	Find/Brow	Group/ung	Group/ung	Import/ex	Import/ex	Save/Load	Save/Load										Connect/D	Connect/D	Send/Rece	Send/Rece							
Action Synonym:	Install | New | De	Drop | Clear | Uni							Open | Backup | R	Backup	Replicate	New | Install | Cre		Remove | Clean | Dele	Update | Change | Gra				Upgrade | Alter | Chang					Mount | Map | Re				Monitor		Audit
ActionNam	Create	Remove	Browse	Find	Group	UnGroup	Export	Import	Load	Save	Copy	Design	View	Clear	Configure	Move	Secure	Tune	Update	Connect	Disconnect	Receive	Send	Link	Compare	Convert	Measure	Audit	Debug	Monitor
Row Labels																														
⊟[Server Collection or Cluster]	y	y	y	y	y	y	y	y	y	y	y	y	y	y	y	y	y	y	y	y	y			y			y	y	y	y
⊟[OS Server - Physical or Virtual]	y	y	y	y	y	y			y	y	y		y		y	y	y	y	y	y	y			y			y	y	y	y
⊟[Database Server Collection]																														
⊟[Database Server Instance]	y	y	y	y	y	y			y	y	y	y	y		y	y	y	y	y	y	y			y			y	y	y	y
⊟[User Database Collection]	y	y	y	y	y	y	y	y	y	y	y	y	y	y	y	y	y		y						y	y	y	y	y	y
⊟[User Database]	y	y	y	y	y	y	y	y	y	y	y	y	y	y	y	y	y	y	y	y	y	y	y	y	y	y	y	y	y	y
⊞[Database Diagrams]																														
⊟[Programmability Objects]	y	y	y	y	y	y	y	y	y	y	y	y	y	y		y	y	y	y					y	y	y	y	y	y	y
⊞[Assemblies]	y	y	y	y	y	y	y	y	y	y	y	y	y	y	y	y	y	y	y						y	y	y	y	y	y
⊞[Database Triggers]	y	y	y	y	y	y	y	y	y	y	y	y	y	y	y	y	y	y	y						y	y	y	y	y	y
⊞[Defaults]	y	y	y	y	y	y	y	y	y	y	y	y	y	y	y	y	y	y	y						y	y	y	y	y	y
⊞[Functions]	y	y	y	y	y	y	y	y	y	y	y	y	y	y	y	y	y	y	y						y	y	y	y	y	y
⊞[Rules]	y	y	y	y	y	y	y	y	y	y	y	y	y	y	y	y	y	y	y						y	y	y	y	y	y
⊞[Stored Procedures]	y	y	y	y	y	y	y	y	y	y	y	y	y	y	y	y	y	y	y							y	y	y		
⊞[Types]	y	y	y	y	y	y	y	y	y	y	y	y	y	y	y	y	y		y						y	y	y	y	y	y
⊟[Security]	y	y	y	y	y	y	y	y	y	y	y	y	y	y	y	y	y		y						y	y	y	y	y	y
⊞[Asymmetric Keys]	y	y	y	y	y	y	y	y	y	y	y	y	y	y	y	y	y		y						y	y	y	y	y	y
⊞[Certificates]	y	y	y	y	y	y	y	y	y	y	y	y	y	y	y	y	y		y						y	y	y	y	y	y
⊞[Roles]	y	y	y	y	y	y	y	y	y	y	y	y	y	y	y	y	y		y						y	y	y	y	y	y

Fig. 3. A part of the Action-Object map

high availability is a *feature area* perspective and associates with database backup, recovery, monitoring, performance management, troubleshooting and other related tasks.

3.2 Task Taxonomy Data Cubes

The raw user task data were transformed into a set of standardized scales. For inconsistent rating scales used by different researchers in earlier data collections, 7-point Likert scale was used. The frequency data were converted to the number of occurrence for a year. Time on task were converted to hours. The raw data were then mapped onto the *task taxonomy* model through assigning each data record to the task hierarchy (if there is no existing record, a new level 1/2/3 task is created based on the hierarchy rule). Then the level 3 task data was mapped to the *action-object map*. In the meantime, the record was also tagged with perspectives. E.g., database backup was tagged with DBA role, Windows Server platform, high availability feature area etc.

After the mapping and tagging were complete, the entire database tables were loaded to SQL Server Analysis Services (SSAS) cubes. Task hierarchy, action-object map, and perspective data were defined as dimensions. The task measures such as ratings of importance, complexity, task frequency, and time on task were defined as measures. The SSAS would then enable us to create pivot tables in MS-Excel to explore the relationship among task measures and task dimensions.

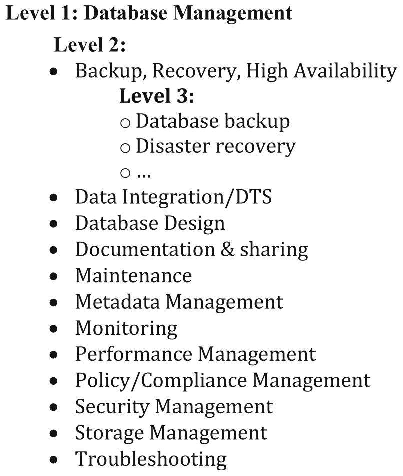
4 Findings

4.1 Database Management Task Hierarchy

A task hierarchy was generated from the SSAS cubes. The level 1 tasks were defined with SQL Server engineering architects and domain experts as four areas of user job roles. As our study participants were mostly DBAs, the data distribution skewed towards more database management tasks (83 % data points), less on application/BI development and management (11 %), service lifecycle management (6 %), and almost none on data consumption (0.1 %). Hence, we focused on database management tasks.

Under database management, twelve responsibility areas were classified as level 2 tasks: backup and recovery, data integration, database design, database maintenance, database monitoring, performance management, security management, storage management, troubleshooting, metadata management, documentation and sharing, and policy/compliance management (see Fig. 4).

Under each level 2 task, numerous level 3 tasks were classified. For example, under backup and recovery, level 3 task could include database backup, disaster recover, database restore, backup monitoring, maintenance plan and etc.



Level 1: Database Management

Level 2:

- Backup, Recovery, High Availability

  Level 3:

  ○ Database backup

  ○ Disaster recovery

  ○ ...

- Data Integration/DTS
- Database Design
- Documentation & sharing
- Maintenance
- Metadata Management
- Monitoring
- Performance Management
- Policy/Compliance Management
- Security Management
- Storage Management
- Troubleshooting

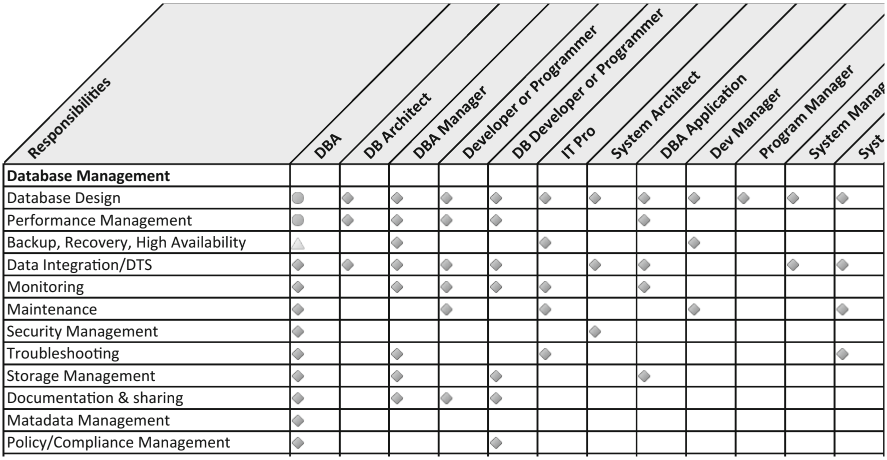
Fig. 4. Example of database management task hierarchy

4.2 Task Hierarchy and Roles

Our early user interviews found that database user roles are heavily overlapped in small businesses with a common notion of "I wear multiple hats – I do it all". In large and enterprise companies, user roles become more distinct, and tiers are created to meet the business workload needs. Research shows that in even in these environments, however, roles expand and change, overlap, get delegated across individuals, and differ greatly from company to company. This has implications for the way we construct our UIs, information architectures, roles and permissions, and delegation models.

Role Differentiation: Fig. 5 shows the responsibility areas that are covered by different roles. "DBA" as a general database role covers almost the entire responsibilities area. However, the general DBA role could split into multiple roles such as Application DBA, System DBA, Data Warehouse DBA, BI DBA in larger firms. The splitting of a big DBA role implies the shift of responsibility focuses - from covering the broad area of responsibilities to focusing on a smaller set of specialized responsibilities. For instance, System DBAs are responsible for troubleshooting, database deployment, and monitoring and maintenance tasks at the Operating System level. Many BI or Data Warehouse DBAs roles are generated (mostly from senior DBAs) in companies where large BI systems are deployed.

Role Stratification: DBAs at different tiers working at different tactical-strategic levels are common in many enterprise companies, especially in companies which have outsourced their tier-1 support. Indirectly, Fig. 5 provides comparative information on role-responsibility mapping among DBAs, DBA managers, and DATABASE Architects. As DBAs move up the ladder from senior DBA to DBA Managers, and to DBA Architects, their responsibilities shift from working at the tactical level to strategic



Responsibilities	DBA	DB Architect	DBA Manager	Developer or Programmer	DB Developer or Programmer	IT Pro	System Architect	DBA Application	Dev Manager	Program Manager	System Manag	Syst
Database Management												
Database Design	●	◆	◆	◆	◆	◆	◆	◆	◆	◆	◆	◆
Performance Management	●	◆	◆	◆	◆			◆				
Backup, Recovery, High Availability	△		◆			◆			◆			
Data Integration/DTS	◆	◆	◆	◆	◆		◆	◆			◆	◆
Monitoring	◆		◆	◆	◆	◆		◆				
Maintenance	◆			◆		◆			◆			◆
Security Management	◆						◆					
Troubleshooting	◆		◆			◆						◆
Storage Management	◆		◆		◆			◆				
Documentation & sharing	◆		◆	◆	◆							
Matadata Management	◆											
Policy/Compliance Management	◆				◆							

Fig. 5. The coverage of responsibilities and tasks by database user roles

tects. As DBAs move up the ladder from senior DBA to DBA Managers, and to DBA Architects, their responsibilities shift from working at the tactical level to strategic planning, design, performance, and data quality and data integration related responsibilities.

4.3 Task Hierarchy and Degree of Relevance

The data cubes provides relevant measures on task hierarchy and perspectives from the task measures collected. The aggregated importance rating and frequency provided a good estimate of how relevant of a lower level task to its higher level task and to a particular perspective such as a job role (see Table 2). For example, under level 2 - performance management, relevant tasks include performance monitoring, performance tuning, profiler tracing, and troubleshooting. This could be further traced down to the action-object level. Relevant objects include performance data, performance plans, queries, stored procedures, user databases, and server instances. Relevant actions include tune, monitor, trace, debug, and check or review (see Table 3).

5 Implications and Applications

5.1 Information Architecture Design Constructs

The task taxonomy model has provided instruments for defining database enterprise system IA. The task hierarchy and its interaction with different perspectives such as job roles reflects the ways how the users structure, slice and dice their work towards achieving different goals. Level 1 tasks represent different job roles and therefore correspond to different tools or applications that are designed for those roles. Level 2 tasks of responsibilities represent the clustering of users' common goals in dealing with the same set of objects in the same context. They are the workspaces inside the tool for a particular user role. The responsibilities are similar to the user goals described in Goal-Directed Design by Cooper et al. [5].

Design Role-Based Workspace as Distinct Interface for Best Fulfilling Responsibilities. In UI design, a responsibility can be supported through a common workspace. For example, database design is a major responsibility for DBAs, DB Architects, and DB Developers. Database objects include servers, databases, tables, and foreign relationships. A database design workspace serves a common goal to monitor, create, and update all those objects. However, roles with the same responsibility may have different focuses. The workspace should be optimized for different tasks for each role. Of course, different roles may have different set of workspaces.

Optimize for Personalized Views to Support Role Differentiation. For database design tasks, DB architects care more about logical designs but DBAs care about how to implement it. DB architects are required of a deep understanding of business requirements, technology, and resources to create an appropriate logical design. In contrast, DBAs may care more about the physical implementation. Therefore, DB Architects and DBAs need different focal views when dealing with different forms of database design.

Table 2. Examples of tasks and task measures for degree of relevance

Task hierarchy	Importance	Frequency	Time on task	Satisfaction
Level 1 - Database Management	**6.30**	**3.79**	**6.96**	**4.73**
Level 2 - Policy/Compliance Management	**7.00**	**5.00**	**2.00**	**1.00**
Level 2 - Backup, Recovery, High Availability	**6.83**	**3.76**	**7.48**	**5.39**
Level 3 - Monitoring Backups	7.00	4.80	1.67	6.00
Level 3 - Test Or Check Backup	7.00	4.67	4.00	5.50
Level 3 - Review Backup/Recovery	7.00	3.50	2.25	3.25
Level 3 - Test Failover	7.00	3.00	3.00	7.00
Level 3 - Test Recoverability	7.00	3.00	3.00	3.63
Level 3 - Database Recovery	7.00	2.50	3.17	5.50
Level 3 - Log Shipping	7.00	2.00	6.00	5.50
Level 3 - Database Restore	6.86	3.55	2.17	5.00
Level 3 - Database Backup	6.84	4.35	1.80	5.45
Level 3 - Setup Backup/Maintenance Plan	6.83	2.78	28.56	6.00
Level 3 - Disaster Recovery	6.67	3.67	15.50	6.50
Level 3 - HA Maintenance	6.63	1.00	80.00	4.00
Level 3 - Maintain Backup Restore	4.00	4.00	0.17	5.50
Level 2 - Monitoring	**6.65**	**4.23**	**2.38**	**4.81**
Level 2 - Security Management	**6.58**	**3.53**	**1.15**	**3.86**
Level 2 - Maintenance	**6.57**	**4.74**	**1.88**	**5.44**
Level 2 - Data Integration	**6.45**	**3.65**	**6.05**	**4.36**
Level 2 - Database Design	**6.37**	**3.82**	**11.13**	**4.70**
Level 2 - Documentation & Sharing	**6.20**	**3.33**	**6.59**	**3.68**
Level 2 - Performance Management	**5.92**	**3.69**	**5.48**	**4.78**
Level 2 - Troubleshooting	**5.40**	**3.82**	**2.64**	**4.26**

Provide Relevant Task and Actions in the Context. In database administration, team communication, project versioning, documentation, guidelines, script library, and templates are critical for the job. These should be designed in the context of the corresponding workspace and tasks.

Balance the Focal Views and Context Using Relevance Measures. The task measures provide comparative weights for any workspace, tasks, and UI objects and actions. In performance monitoring workspace, database availability and response latency are the primary goals. They should be designed as the focal view whereas performance guidelines should be minimized in its context.

Table 3. Example of relevant tasks, objects and actions under L1 database management

L2 Responsibility	L3 Relevant tasks	Relevant objects	Relevant actions
Performance management	Performance monitoring	Performance data	Tune
	Performance tuning	Performance plan	Monitor
	Profiler tracing	Stored Procedure or query	Trace
	Troubleshooting	Server instance	Debug or troubleshoot
	Tuning codes	User databases	Check or review
	Maintain performance	Database server	Create
	Performance	Views	Design
	assessment	Indexes	Maintain
	Query performance	Programmability	Find
	File management	objects	Test
		Tables	Upgrade
		Design, models	Measure
			Run
			View

5.2 An Extended Framework to Goal-Directed Design

Goal-Directed Design [5] has a simple premise: "If we design and construct products in such a way that the people who use them achieve their goals, these people will be satisfied, effective, and happy and will gladly pay …, translate into business success" (page 3). It differentiates goals from tasks or activities that "a goal is an expectation of an end condition, whereas both activities and tasks are intermediate steps…" and goals are "driven by human motivation, they change very slowly … over time" (page 15). Goal-Directed Design draws a direct line between a user's initial state and the end state of the objectives he or she wants to achieve, eliminating the noise from other aspects such as activities, tasks, technologies, or solutions.

When come to IA design, the task taxonomy model leads us to focus on user goals using quantitative data – inducing user goals through the task hierarchy, action-object map, with relevance measures, and placing the user goals in the multi-user, multi-group, multi-purpose, and collaborative context. As shown in Fig. 6, the level 2 responsibilities in the task hierarchy represent clusters of users' common goals and should be designed as workspaces in the UI. The level 3 tasks represent the individual's long term goals within that responsibility. Therefore, when a large amount of user task data were collected and degree of relevance of each task were calculated, the workspaces could be organized and personalized as high level navigation nodes for each user role to better fulfill that responsibility. Within that workspace, the focal views could be designed as the evolving user needs for the most important and frequent tasks. Many supportive tasks would be designed as its context.

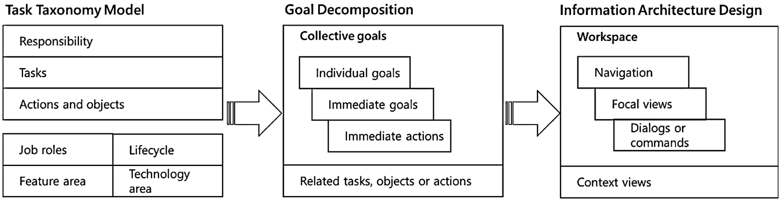
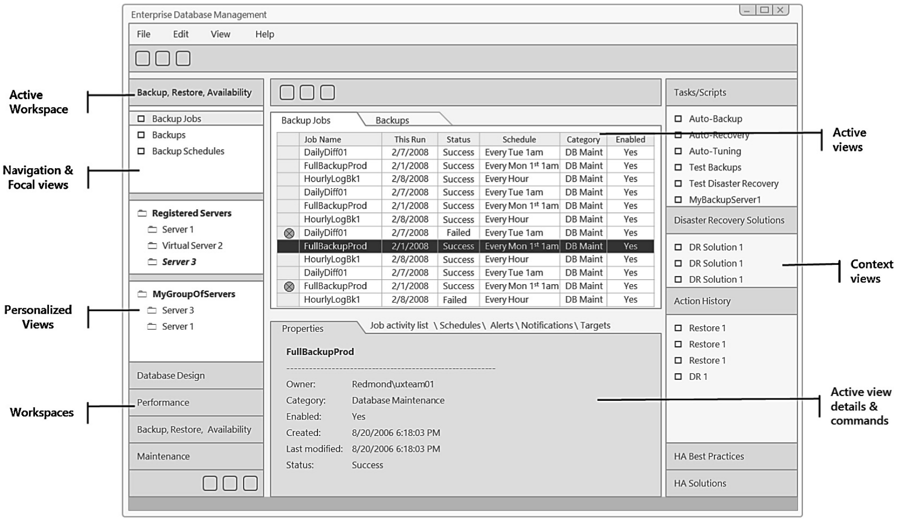


Fig. 6. Extended Framework of applying task taxonomy model in IA design

5.3 Design Illustration

To illustrate the idea, Fig. 7 shows the use of the task data to design a database management IA. For a DBA role such as a production DBA, the enterprise database management tool is customized with fewer workspaces that are optimized for his/her job. Within the Backup and Recovery workspace, a navigation structure with focal views and personalized views provides a quick access to the important backup jobs, backups and schedules. In the context, tasks/scripts, disaster recovery solutions, and others are designed as context views – context sensitive to the active focal view. Switching over to the performance workspace, he/she may see performance monitoring meters, gauges, alerts and recommended actions with backup and recovery tasks in its context.



	Job Name	This Run	Status	Schedule	Category	Enabled
	DailyDiff01	2/7/2008	Success	Every Tue 1am	DB Maint	Yes
	FullBackupProd	2/1/2008	Success	Every Mon 1st 1am	DB Maint	Yes
	HourlyLogBk1	2/8/2008	Success	Every Hour	DB Maint	Yes
	DailyDiff01	2/7/2008	Success	Every Tue 1am	DB Maint	Yes
	FullBackupProd	2/1/2008	Success	Every Mon 1st 1am	DB Maint	Yes
	HourlyLogBk1	2/8/2008	Success	Every Hour	DB Maint	Yes
⊗	DailyDiff01	2/7/2008	Failed	Every Tue 1am	DB Maint	Yes
	FullBackupProd	2/1/2008	Success	Every Mon 1st 1am	DB Maint	Yes
	HourlyLogBk1	2/8/2008	Success	Every Hour	DB Maint	Yes
	DailyDiff01	2/7/2008	Success	Every Tue 1am	DB Maint	Yes
⊗	FullBackupProd	2/1/2008	Success	Every Mon 1st 1am	DB Maint	Yes
	HourlyLogBk1	2/8/2008	Failed	Every Hour	DB Maint	Yes

FullBackupProd

Owner:	Redmond\uxteam01
Category:	Database Maintenance
Enabled:	Yes
Created:	8/20/2006 6:18:03 PM
Last modified:	8/20/2006 6:18:03 PM
Status:	Success

Fig. 7. Illustration of applying task taxonomy model and data to enterprise database management tool design. Note: the design was only an illustration of the extended task taxonomy design framework. It was not designed for any potential or future Microsoft products.

6 Summary

From features-driven to Goals-Directed design, from easy-to-make to easy-to-use, the shift into a user centered design culture is critical to user satisfaction, customer loyalty and product success. Such a shift requires a re-thinking of our product design and development processes, attention to the overall user experience, and the dedication to a comprehensive understanding of user's responsibilities, tasks and organizational context, as well as their personality and emotion aspects if possible.

In this project, the task taxonomy model and task data provide clear underpinnings for planning and designing an enterprise IA, and for sorting out the complex relationships among individual and team goals and responsibilities. It is an extension and quantification of Goal-Directed Design. Placing users in a goal-structured workspace can reduce the costs and shorten the time required for training during role shifts and transitions, and drive team collaboration. Our recent dashboard and role-based designs on enterprise management tools have shown greater usability and higher customer satisfaction – partially attributing to this project.

Acknowledgements. I'd like thank Dave Campbell, Dave Nettleton and Shawn Bice for project sponsorship, Mark Stempski, George Engelbeck, Candace Soderston, Lisa Mueller, Aimee Freeding, Nate Gunderson and Buck Woody for data collection and model discussions.

References

1. Wyllie, J.: Taxonomies: Frameworks for Corporate Knowledge. Ark Group, in association with Inside Knowledge, London (2003)
2. Arevuo, M.: Book Review: Taxonomies: Frameworks for Corporate Knowledge by Jan Wyllie. IK Magazine **7**(10), Posted 20 July 2004
3. Malafsky, G.: Knowledge taxonomy. http://wiki.nasa.gov/federal-knowledge-management-working-group-kmwg/files/2013/06/Knowledge_Taxonomy.pdf
4. Kwasnik, B.H.: The role of classification in knowledge representation and discovery. Libr. Trends **48**(1), 22–47 (1999)
5. Cooper, A., Reimann, R., Cronin, D.: About Face 3: The Essentials of Interaction Design. Wiley, New York (2007)