Acta Universitaria
DIRECCIÓN DE APOYO A LA INVESTIGACIÓN Y AL POSGRADO

A Comparative Analysis of Task Modeling Notations.

Josefina Guerrero-García*, Juan González-Calleros* , Jean Vanderdonckt**

ABSTRACT

Task Models describe how to perform activities to reach users' goals. Task models represent the intersection between user interface design and more systematic approaches. Task models can be represented at various abstraction levels. When designers want to specify only requirements regarding how activities should be performed, they consider only the main high-level tasks. On the other hand, when designers aim to provide precise design indications then the activities are represented at a small granularity, thus including aspects related to the dialogue model of a user interface (which defines how system and user actions can be sequenced). In this paper a comparative analysis of selected models involving multiple users in an interaction is provided in order to identify concepts which are underexplored in today's multi-user interaction task modeling. This comparative analysis is based on three families of criteria: information criteria, conceptual coverage, and expressiveness. Merging the meta-models of the selected models enables to come up with a broader meta-model that could be instantiated in most situations involving multi-user interaction, like workflow information systems, CSCW.

RESUMEN

Entender la forma de cómo realizar el trabajo en las organizaciones es parte importante para alcanzar los objetivos, los modelos de tareas ayudan a realizar esta tarea. Estos modelos son la intersección entre el diseño de la interfaz de usuario y enfoques sistemáticos. Existen diferentes niveles de abstracción para representar un modelo de tarea. Cuando los diseñadores requieren sólo especificar los requisitos con respecto a cómo las actividades se deben realizar, consideran sólo las tareas de alto nivel. Por otro lado, cuando los diseñadores tienen por objeto proporcionar indicaciones precisas entonces las actividades están representadas con mayor granularidad, incluidos los aspectos relacionados con el modelo de diálogo de una interfaz de usuario (que define cómo las acciones del sistema y el usuario pueden ser secuenciados). En este artículo se presenta un análisis comparativo de modelos de tareas con el fin de identificar conceptos comunes. Este análisis comparativo se basa en tres criterios: información, la cobertura conceptual y expresividad. La fusión de los resultados derivó en un meta-modelo robusto que cubre la mayoría de las situaciones concernientes al modelado de tareas, incluyendo: la interacción de varios usuarios, los sistemas de información, el flujo de trabajo, y CSCW.

Recibido: 9 de Enero de 2012
Aceptado: 14 de Febrero de 2012

INTRODUCTION

In Human-Computer Interaction (HCI) and User Interface (UI) design, many roles have been investigated with respect to the multiplicity of the entities playing a role in these models: multiple computing platforms, multiple channels, multiple interaction techniques, multiple modalities, multiple environments, and multiple users.

In particular, multi-target UIs explore variations of multiple contexts of use where the context of use is understood as a user interacting with a computing platform in a given environment. Therefore, multiple contexts of use necessarily mean multiple variations of these three dimensions. Among these dimensions, the multiplicity of users has been less researched than the others and has been investigated in different domains ranging from HCI, Computer-Supported Collaborative Work (CSCW) to Collaborative Systems and Workflows. Multi-user interaction is hereby referred to as a context of use where multiple users are initiating some interaction and/or receiving the feedback of some

Palabras clave:
Componente; patrones de diseño; lenguaje de descripción de interfaces de usuario; flujo de interfaces de usuario.
Keywords:
Component; design pattern; user interface description language; user interface flow.

*Universidad Autónoma de Puebla, 14 Sur y Av. San Claudio, Edificio 136, Ciudad Universitaria (CU), C. P. 72570 , Puebla, Mexico. Email: {jguerrero,juan.gonzalez}@cs.buap.mx.
**Louvain Interaction Lab, Louvain School of Management, Université Catholique de Louvain, Place des Doyens I, B-1348, Louvain-la-Neuve, Belgium. Email: jean.vanderdonckt@uclouvain.be

previously existing interaction, perhaps in multiple environments. Multi-user interaction is significant in a certain amount of areas such as: any circumstances where multiple users are involved, whether they are located in the same environment or not (e.g., collaboration, cooperation, competition, and coopetition), where several users are networked in a workflow, where they have individual or shared tasks, where the tasks are multi-user by nature. The problem is that these areas all have their respective understanding and definition of multiple users involved in an interaction. This situation leads to a series of important shortcomings, among them are:

- **Lack of understanding:** the basic concepts of multi-user interaction modeling are not always well mastered and properly understood, such as the rationale behind their method, their entities, their relationships, their vocabularies, and the intellectual operations involved for modeling these aspects.
- **Matching concepts** across two different models or more is difficult. It is even likely that sometimes no matching across these concepts could be established.
- **Communication among designers is reduced** : due to the lack of software interoperability, a designer may experience some trouble in communicating the results of a multi-user interaction model to another stakeholder of the UI development team. In addition, any transition between persons may generate inconsistencies, errors, misunderstandings, or inappropriate modeling.
- **Heterogeneousness:** these concepts, as they were initiated by various methods issued from various disciplines, are largely heterogeneous.
- **Lack of software interoperability:** since model-based tools do not necessarily share a common format, they are only restricted to those models which are expressed according to their own, possibly proprietary, format.
- **Duplication of research and development efforts:** due to the aforementioned differences, different research and development teams may reproduce similar efforts but towards their own format and terminology, thus reducing significantly the ability to raise incremental research.

This shortcoming is particularly important for software development efforts which are resource-consuming. To address the above shortcomings, we assigned ourselves the next goals:

1. To provide an improved conceptual and methodological understanding of the most significant models involving multiple users and their related concepts.
2. To establish semantic mappings between the different models so as to create a transversal understanding of their underlying concepts independently of their peculiarities. This goal involves many activities such as vocabulary translation, expressiveness analysis, identification of degree of details, identification of concepts, and emergence of transversal concepts.
3. To rely on these semantic mappings to develop a multi-user model editor that accommodates any type of input. This editor should help designers and developers to derive UIs for these multiple users independently of the underlying model. The ultimate goal is to capitalize design knowledge into a single tool and to avoid reproducing identical development effort for each individual model.

The remainder of this paper is structured as follows: In the remaining of the paper we present an overview of select models, thus establishing a comparative analysis and the results provided in order to propose a meta-model gathering the concepts identified. Following this, a case study and a tool supporting the meta-model are presented. The paper is wrapped up by summarizing our work, deriving conclusions and addressing future work and challenges.

BRIEF OVERVIEW OF TASK MODELS

A number of task modeling notations have been developed in the human-computer interaction (HCI) communities, often with different goals and different strengths. In this section, we discuss some notations, examining which characteristics they exhibit and which attributes they cover.

In this section, we present the steps of the method followed to build a uniformed task model from a number of existing task models. A *task model* is referred to as any model produced by a specific task analysis method. The *uniforming* of an individual task model is the process of expressing an individual task model into a *uniformed model* in such a way that its concepts have always the same form, manner, or degree, and present an unvaried appearance of concepts. Our method for uniforming task models consists of four major steps: selection of individual task models, identification of the concepts within each model, representation of those concepts into a meta-model, and consolidation of these meta-models into one single meta-model, called *multi-user interaction model*. Even if these steps are presented sequentially, some feedback is still possible at some points. It is probably beyond our capabilities to consider all existing task models. The following criteria were used:

- The task models should be integrated in a development methodology as a core or side component and tool supported.
- The task models should be widespread and accepted within the Human-Computer Interaction (HCI) community.
- The selected models should be supported by theoretical studies to assess their soundness and experimental studies for effective case studies.

After selecting the individual task models, the foundation references of each chosen task model were analyzed. Each model was then decomposed into constituent concepts using an entity-relationship method and UML representation. The terminology used in original references to refer to concepts was preserved. A definition of each concept is then given. For the sake of concision, only relevant definitions of concepts were retained. These concepts are then represented into a task meta-model, which is made up of entities and relationships expressed according to an entity-relationship-attribute methodology. Finally, a multi-user task meta-model is obtained from the task meta-models. To build this final meta-model, different intellectual operations have been performed.

Firstly, a *syntactical uniforming* has been conducted to provide a single way of referring to different concepts where possible. This step implies that concepts having the same definition but different names were uniformed under a same label. For concepts having different definitions, even if they refer to a similar fundamental concept, a *semantic uniforming* is needed. This step implies the identification of semantic mappings between concepts having different aims and scopes. To maximize the semantic scope of the uniformed task meta-model, the *union* of the concepts present in each particular task meta-models was preferred rather than the *intersection*. Indeed, choosing the intersection would produce an "emergent kernel of concepts" common to all methods, but this set may be rather limited. Conversely, the union while keeping commonalities preserves specific contributions of individual models. In order to avoid the problem of an all embracing model, some concepts (i.e., entities, relationships, or attributes) were withdrawn from this union for several reasons: the concept is semantically redundant with an already existing concept, the concept is not practically used by the methodology in which the particular task model is defined, the concept does not basically belong to the task model but rather to other models like user model, organization model, domain model, or presentation model. This reason is motivated by the *Separation of concern* principle which assumes that only concepts relevant to a similar domain of discourse should be kept in a particular model, thus avoiding mixing different concepts into a single model.

The Task modelling notations reviewed are: AMBOSS [12]; ANSI/CEA-2018 [10], ConcurTaskTrees (CTT) [24], Diane+ [29], GOMS [18] [16], Groupware Task Analysis (GTA) [1], Hierarchical Task Analysis (HTA) was developed by [3] [23] [1], Task Knowledge Structure (TKS) method [15] [17] [25], Task Object-Oriented Description (TOOD) [19] [28], and Interface eXtensible Language (UsiXML) [31] [14]. Due to space reasons we discuss in detail one notation to illustrate the findings that area summarized in figure 2.

ANSI/CEA-2018

ANSI/CEA-2018 is a standard for task model descriptions, which has the potential of significantly improving the usability of computer-controlled electronic products and software interfaces in general. An ANSI/CEA-2018 task model description is an XML document (it is not a graphical formalism) whose syntax and semantics is specified by the standard. The primary use of the XML document is to be interpreted by a device at run-time to guide the user in achieving the described tasks [10]. The key representational features are: Tasks, Input and output parameters, User intent concept, Preconditions and postconditions, Task decomposition, Binding, Grounding, Temporal order, and Applicability conditions.

The concept of *task* (also called activity, goal, job, action) is at the heart of the standard. Tasks vary widely in their time extent, and some have unbounded time extent. Tasks typically involve both human participants and electronic devices. Some tasks may be performable only by a human being; others may be performed only by an electronic device. Tasks also vary along an abstraction spectrum from what might be called high-level to low-level. A task model defines task *classes*. A task *instance* corresponds to an actual or hypothetical occurrence of a task. Input and output parameters represent the data to be communicated with other tasks. The *input* parameters of a task class should include all data which affects the execution of task instances. The *output* parameters of a task class should include all data which is modified or created during execution of task instances.

A *user intent concept* is a case frame, consisting of a verb and a set of semantic roles of specified types. The following semantic roles are predefined: agent: Agents are entities that bring about a state of affairs; theme: The theme is whatever is acted upon or most affected or undergoes motion of some sort, including motion in a metaphorical sense; instrument: The instrument is whatever is being used to perform the action; and location: Locations are places; they can also serve as the endpoints of paths.

The *precondition* of a task is a partial Boolean function which tests whether or not it is appropriate to perform the task at the moment. The *postcondition* of a task is a partial Boolean function which tests whether a just executed task was successful. A task can be *decomposed* into subtasks which are described as steps for executing the task (hierarchy). The *temporal order* between theses subtasks is by default linear (totally ordered), but ANSI/CEA-2018 also supports the specification of partial orders. The data flow between these steps is specified by the *binding* elements in the subtasks definition. In ANSI/CEA-2018, a binding is equality between an input slot of a decomposition step or an output slot of a decomposition goal, and the value of a function with arguments corresponding either to the output slots of steps or the input slot of the goal. *Grounding* is the binding of primitive tasks (those that do not have subtasks) to a script (written in ECMAScript). In ANSI/CEA-2018, a script is an ECMAScript program which may be associated with one or more tasks classes, platforms and device types and whose properties include an applicability condition. For each decomposition may optionally include an *applicability condition*, which can help the system choose the appropriate decomposition when there is more than one.

A MULTI-USER INTERACTION META-MODEL

The meta-model depicting all these concepts in shown in figure 1. We introduced the notion of color to explicitly show those models whose in-stances are potentially highly dynamic (instances of red classes) at run-time. By run-time is meant when the WfIS is in execution. While all concepts are involved during the design of the WfIS, instances of classes are not necessarily modified or used at run-time. For instance, once a task model relationship has been established during the design of the WfIS there is no way to change it at run-time (green classes). No need to focus on this aspect when considering the implementation of the WfIS.

The color distinction proved to be useful for the implementation of a workflow editor, to discard classes that were not needed at all for the design of the WfIS (red classes), and to keep an understanding of those concepts that at run-time are to be implemented (yellow and red classes).

Notice that there some classes that are both used at design-time and run-time. An instance of this class can change significantly (the agenda is an example as it changes constantly) thus the use of the red color. An instance of a class that changes moderately (the job definition for the execution of the task is an example as the definition does not normally change on a regular basis) thus the use of the yellow color.

Task operators

The task models exhibit a variety of concepts and relationships. The differences between concepts include differences of vocabulary used for the same concept across models. They have, also, different bases of formalization, and scopes. The table 1 briefly illustrates the variations between task models. The comparison is based on *formalization* (this dimension specifies whether a model is based on a formal system or not), *role* (in order to know if the model uses a role concept. Roles are played by agents and are assigned according to organizational rules.), *goal* (some models make a differentiation between tasks and goals), *cooperative aspect* (how it supports cooperative work), *scope of constructors* (expresses the scope of the task elements on which the temporal operators work. The scope can be the *parent* or the *sibling* when any temporal operator constraint affects the ordering, respectively, between a father node in the task decomposition and its children or between siblings of the same father), *decomposition* (show the level of decomposition allowed in the model), *operational level* (the task decomposition level where actions take place), *tools* (tools that have as basis the task model).

Task operators identified in most task modeling notations are [14]: Decomposition relationships, enabling to represent a hierarchical structure of the task tree. Temporal relationships represent a specification of temporal relationships between tasks. They can be *binary* or *unary*.

Binary relationships are a type of temporal relationships that connects several instances of two different tasks. Enabling relationships specify that a target task cannot begin until source task is finished. Disabling relationships refer to source task that is completely interrupted by a target task. Suspend Resume relationships refer to source task that can be partially interrupted by a target task and after the target task is completed the source task will be concluded.

Order Independence relationships are when two tasks are independent of the order of execution. Concurrency with Information Passing relationships are a type of temporal relationships where two tasks are in concurrency execution and passing information between them. Independent Concurrency relationships are a type of temporal relationships where two tasks are executed concurrency but are independent one to each other and there is no information interchange. Enabling with Information Passing relationships specify that a target task cannot be performed until the source task is performed, and that information produced by the source task is used as an input for the target task. Cooperation relationships specify the relationship of cooperation between two or more tasks.

Inclusive Choice relationships specify two tasks that: both could be executed or just one of them or neither of them. Deterministic Choice relationships refer to two source tasks that could be executed but once that one task is initiated the other cannot be accomplished anymore. Undeterministic Choice relationships define the relation between two source tasks in which both task could be started but once one task is finished the other cannot be accomplished anymore. Disabling with Information Passing relationships occur if one task is completely interrupted by another task; and the information produced in the first task is used as an input for the second task.

Unary Relationships are temporal relationships that connect several instances of the same task. Optional relationships refer to source task that are optional.

Iteration relationships indicate source tasks that may be iterated. Finite Iteration tasks indicate tasks that may be iterated n times.

In order to represent group's requirements to coordinate their work among themselves by relying on implicit (e.g., manual, verbal, informal) communication schemes, it is necessary to addressing Mandviwalla & Olfman [21] criteria for support group interactions, such as the following ones we selected in our work: support carrying out group tasks, Support multiple ways to support a group task, support the group evolution over time.

Figure 1 illustrates the concepts that are used to build a multi-user interaction model. Tasks are organized into processes. A task is decomposed into subtasks and operators are used to link them on the same level of decomposition. A task may manipulate objects through actions. We introduce the concept of *Job* instead of *role*. Jobs are the total collection of tasks, duties, and responsibilities assigned to one or more positions which require work of the same nature and level. An organizational unit is a formal group of people working together with one or more shared goals or objectives. It could be composed of other organizational units. Resources are characterized thanks to the notion of user stereotype. But a same task could require other types of resources such as material resources (e.g., hardware, network) or immaterial resources (e.g., electricity, power). The agenda is a list of tasks that are assigned to user stereotype. A user stereotype has one and only one agenda and an agenda belongs to one and only one user stereotype. The concept of agenda is useful to cope with the cooperative aspects. We can allocate or offer tasks to user stereotypes through the agenda.

CONCLUSION

In the research literature there is a wide variety of task models with different approaches, it is difficult to consider all in order to elaborate a comparative analysis. To generate our meta-model, we consider those that are supported by theoretical studies, accepted within the Human-Computer Interaction community, and are integrated in a development methodology.

Task models analyzed in previous sections show a variety of concepts and relationships. Differences between concepts are both syntactic and semantic. Syntactic differences cover differences of vocabulary used for a same concept across models. Semantic differences are related to the conceptual variations across models. Semantic differences can be of major or of minor importance. A major difference consists in the variation of entities or relationships definitions and coverage; for instance, a same concept does not preserve a consistent definition across models. A minor difference consists in the variation of expressing an entity or a relationship. For example, constructors in GTA or TKS express temporal relationship between a task and its subtasks, although the set of constructors is not identical in all models, while operators in CTT are used between sibling tasks. After the analysis of those task models, a multi-users interaction meta-model was generated in order to cover the principal characteristics required to work with multiplicity entities playing a role. The meta-model applies to identify how tasks are structured, who perform them, what their relative order is, how they are offered or assigned, and how tasks are being tracked. Moreover, an editor was developed to put in practice the aforementioned model.

Our meta-model tries to cover the principal aspect required to support group work, it include process, tasks, task operators (including collaboration rela tionship), actions, objects, resources, groups (as an attribute), organizational units, jobs, agendas, goals and rules (both of them as attributes).

In a future work, we would like to integrate in our comparative analysis other task models that are focused on multi-users interaction. Also, it would be interested to integrate a task analysis part, until now our meta-model is devoted to task modeling

ACKNOWLEDGMENT

Josefina Guerrero García and Juan Manuel González Calleros acknowledge CONACYT program "Repatriación" to support this research.

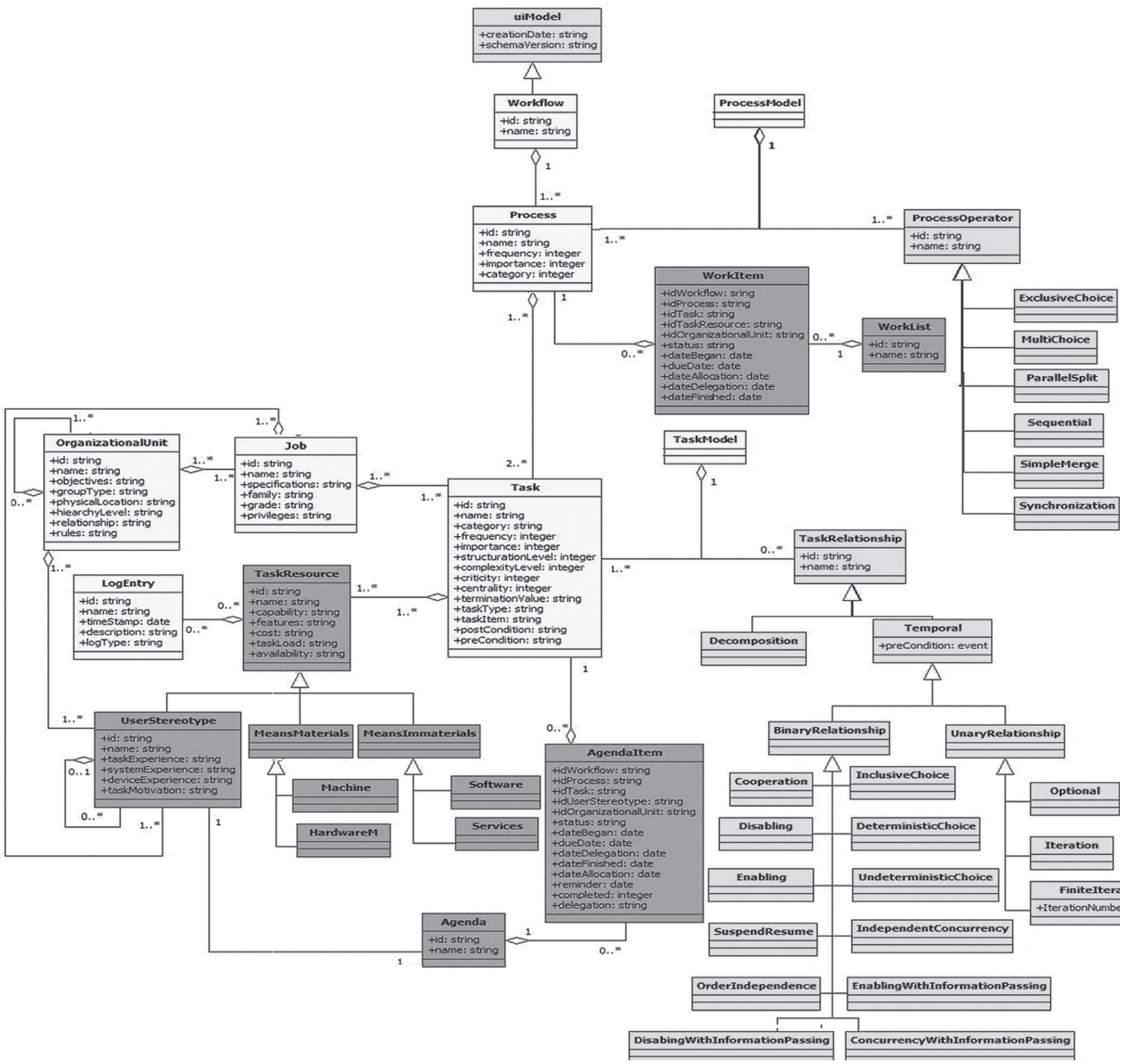

Figure 1. Multi-User Interaction Meta-Model

	AMBOSS	ANSI/CEA	CTT	Diane +	GOMS	GTA	HTA	TKS	TOOD	UsiXML
Decomposition	√ Hierarchy	√ Hierarchy	√ Hierarchy	√ Hierarchy	√ Hierarchy	√ Hierarchy	√ Hierarchy	√ Hierarchy	√ Hierarchy	√ Hierarchy
Sequence	√ SEQ	+- Ordered = true, information passing (Postcondition)	√ Enabling, enabling with information passing	√ Ordered sequence	√ Sequence	√ Seq	√ Fixed sequence	√ Sequence	√ Sequence	√ Enabling, enabling with information passing
Iteration	X	+- MinOccurs+ MaxOccurs	√ Iteration, finite iteration	√ Loop	+- Loop (If, then, else)	X	+- Stop rules	X	X	√ Iteration, finite iteration
Choice	√ ALT	+- Precodition	√ Choice	√ Required choice, free choice	+- Or (If, then, else)	√ Or	√ Selective rule	√ Or	√ Choice	√ Deterministic choice, undeterministic choice, inclusive choice
Optionality	+- Barrier	+- MinOccurs+ MaxOccurs	√ Optional	√ Optional	+- Optional (If, then, else)	+- Start condition	X	X	X	√ Optional
Interruption	X	X	√ Suspend-resume, disabling	X	+- Interruption (If, then, else)	+- Stop condition	√ Stop rules	X	√ Interruption	√ Suspend- resume, disabling, disabling with information passing
Concurrency	√ SER	+- Ordered = false	√ Concurrent, concurrent communicating tasks, independence	√ unordered sequence	+- Concurrency (If, then, else)	X	+- Selective rule	X	√ Concurrency	√ Independent concurrency, concurrency with information passing, order independence
Cooperation	+- Precondition	X	√ Cooperative	X	X	√ Cooperation	+- Teamwork	√ Collaboration(FKS extension)	√ Collaboration	√ Cooperation
Parallel	√ PAR, SIM	X	X	√ Parallel	X	√ And	√ Dual task (time sharing)	√ And	√ Simultaneity	√ parallelSplit (process model)

√ Explicit supported, +- Implicit supported, X Not specified

Figure 2 . Task Modeling Operators

REFERENCES

[1] Annett, J. (2004) Hierarchical Task Analysis. *The Handbook of Task Analysis for Human-Computer Interaction,* pp. 67-82.

[2] Annett, J., Cunningham, D., Mathias-Jones, P.(2000). A method for measuring team skills. *Ergonomics,* 43:1076-1094.

[3] Annett, J., Duncan, K.(1967). Task analysis and training design. *Occupational Psychology,* 41(2):11-227.

[4] Beard, D. V., Smith, D. K., Denelsbeck, K. M. (1996). QGOMS: A direct-manipulation tool for simple GOMS models. *Proceedings of ACM Conference on Human Factors in Computing Systems (CHI '96),* 2:25-26.

[5] Baumeister, L. K., John, B. E., Byrne, M. D. (2000). A comparison of tools for building GOMS models: Tools for design. *Proceedings of ACM Conference on Human Factors in Computing Systems (CHI 2000),* 1:502-509.

[6] Bolognesi, T. and Brinksma, E. (1987). Introduction to the ISO Specification Language LOTOS. *Computer Networks and ISDN Systems,* 14(1):25-59.

[7] Bomsdorf, B. and Swillius. (1998). From task to dialogue: Task based user interface design. *SIGCHI Bulletin,* 30(4):40-42.

[8] Calvary, G., Coutaz, J., Thevenin, D., Limbourg, Q., Bouillon, L., Vanderdonckt, J. (2003). A Unifying Reference Framework for Multi-Target User Interfaces. *Interacting with Computers,* 15(3):289-308.

[9] Card, S.K., Moran, T.P. and Newell, A. (1983). *The Psychology of Human-Computer Interaction.* Lawrence Erlbaum Associates.

[10] CEA-2018 (ANSI)().Task Model Description (CE TASK 1.0). CEA. [CEA 2007]. http://www.ce.org/Standards/browseByCommittee_4467.asp.

[11] Garrido Bullejos, J.L. (2003). *AMENITIES: Una metodología para el desarrollo de sistemas cooperativos basada en modelos de comportamiento y tareas. Ph.D thesis.*

[12] Giese, M., Mistrzyk, T., Pfau, A., Szwillus, G. and von Detten, M. (2008). AMBOSS: A Task Modelling Approach for Safety-Critical Systems.To appear in *Proc. of 7th Int. Workshop on TAsk MOdels and DIAgrams TAMODIA'2008.* Lecture Notes in Computer Science, Springer-Verlag.

[13] Guerrero, J., Vanderdonckt, J., Lemaigre, Ch.,González Calleros, J.M. (2008). How to Describe Workflow Information Systems to Support Business Process. *10th IEEE Conference on E-Commerce Technology and the Fifth IEEE Conference on Enterprise Computing.*

[14] Guerrero, J., Vanderdonckt, J., Gonzalez Calleros, J.M. (2008). FlowiXML: a Step towards Designing Workflow Management Systems.*Journal of Web Engineering,* 4(2):163-182.

[15] Johnson, P. and Johnson, H. (1989). Knowledge analysis of task: Task analysis and specification for human-computer systems. *Engineering the human-computre interface,* A. Downton, (Ed.) pp.119-144.

[16] John, B. E., Kieras, D. E. (1996). The GOMS family of user interface analysis techniques: Comparison and contrast. *ACM Transactions on Computer-Human Interaction,* 3:320-351.

[17] Johnson, P., Hyde, J. (2003). Towards Modeling Individual and Collaborative Construction of Jigsaws Using Task Knowledge Structures (TKS). *ACM ToCHI,* 10(4):339-387.

[18] Kieras, D. (2004). GOMS Model for Task Analysis. *The Handbook of Task Analysis for Human-Computer Interaction,* pp.83-116.

[19] Mahfoudhi, A. (1997). *TOOD: Une méthodologie de description orientée objet des tâches utilisateur pour la spécification et la conception des interfaces hommes-machines. Ph.D. thesis,* Univ. Of Valenciennes.

[20] Mahfoudhi, A., Abed, M., Tabary,D. (2001). From the formal specifications of user tasks to the automatic generation of the HCI specifications. *People and computers XV,* pp.331-347.

[21] Mandviwalla, M., Olfman, L. (1994). What do groups need? A proposed set of generic groupware requirements.*ACM Transactions on Computer-Human Interaction,* 1(3):245-268.

[22] Montero, F., López-Jaquero, V. (2006). IdealXML: An Interaction Design Tool-A Task-Based Approach to User Interfaces Design . *Proc. of 6th Int. Conf. on Computer-Aided Design of User Interfaces CADUI'2006,* Chapter 20, pp.245-252.

[23] Ormerod, T.C., Shepherd, A. (2004). Using Task Analysis for Information Requirements Specification: The Sub-Goal Template (SGT) Method. *The Handbook of Task Analysis for Human-Computer Interaction,* pp.347-365.

[24] Paternò, F. (1999). *Model Based Design and Evaluation of Interactive Applications.* Springer Verlag, Berlin

[25] Pinelle, D., Gutwing, C. (2001). Group task analysis for groupware usability evaluation.*Proc. IEEE WetIce* .

[26] Rubart, J., Dawabi, P. (2004). Shared data modeling with UML-G. *International Journal of Computer Applications in Technology,* 19(3/4):231-243.

[27] Russell, N., van der Aalst, W.M.P., ter Hofstede, A.H.M. and Edmond, D. (2005). Workflow Resource Patterns. *7th Conference on Advanced Information Systems Engineering (CAISE'05).*

[28] Tabary, D., Abed, M. (2000). A Software Tool Task Object-Oriented Design (TOOD) Method. *Proceedings ACIDCA'2000*

[29] Tarby, J., Barthet, M. (1996). The DIANE+ Method. *International Workshop of Computer-Aided Design of User Interfaces.*

[30] Tauber, M.J. (1990). ETAG: Extended task action grammar, a language for the description of the user's task language. *Proceedings of the 3rd IFIP TC 13 Conference On Human-Computer Interaction Interact 90,* pp.163-168,.

[31] UsiXML. *USer Interface eXtensible Markup Language.* http://www.usixml.org.

[32] van der Aalst, W., van Hee, K. (2002). *Workflow Management: Models, Methods, and Systems* . THE MIT Press, Cambridge.

[33] van der Veer, G.C., van der Lenting, B.F., Bergevoet, B.A.J. (1996). GTA: Groupware Task Analysis - Modeling Complexity. *Acta Psychologica,* 91:297-322.

[34] van Welie, M., van der Veer, G.C., Eliëns, A. (1998). EUTREPE: Tool support for analyzing cooperative environments. *Ninth European Conference on Cognitive Ergonomics.*

[35] van Welie, M. (2000). *Task-Based User Interface Design.* SIKS Dissertation Series.