



HAL
open science

Extending Procedural Task Models by Explicit and Systematic Integration of Objects, Knowledge and Information

Célia Martinie, Philippe Palanque, Martina Ragosta, Racim Fahssi

► **To cite this version:**

Célia Martinie, Philippe Palanque, Martina Ragosta, Racim Fahssi. Extending Procedural Task Models by Explicit and Systematic Integration of Objects, Knowledge and Information. 31st European Conference on Cognitive Ergonomics (ECCE 2013), ACM Special Interest Group on Computer-Human Interaction, Aug 2013, Toulouse, France. pp.1-10, 10.1145/2501907.2501954 . hal-04083401

HAL Id: hal-04083401

<https://hal.science/hal-04083401>

Submitted on 27 Apr 2023

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Open Archive TOULOUSE Archive Ouverte (OATAO)

OATAO is an open access repository that collects the work of Toulouse researchers and makes it freely available over the web where possible.

This is an author-deposited version published in : <http://oatao.univ-toulouse.fr/>
Eprints ID : 12675

Official URL: <http://dx.doi.org/10.1145/2501907.2501954>

To cite this version : Martinie De Almeida, Celia and Palanque, Philippe and Ragosta, Martina and Fahssi, Racim Mehdi *Extending Procedural Task Models by Explicit and Systematic Integration of Objects, Knowledge and Information*. (2013) In: 31st European Conference on Cognitive Ergonomics (ECCE 2013), 26 August 2013 - 28 August 2013 (Toulouse, France).

Any correspondance concerning this service should be sent to the repository administrator: staff-oatao@listes-diff.inp-toulouse.fr

Extending Procedural Task Models by Systematic Explicit Integration of Objects, Knowledge and Information

Célia Martinie, Philippe Palanque

IRIT, University of Toulouse
118, route de Narbonne
31042 Toulouse Cedex 9, France.
{martinie, Palanque}@irit.fr

Martina Ragosta

Deep Blue
Piazza Buenos Aires 20
00198 Roma, Italy.
martina.ragosta@dblue.it

Racim Fahssi

IRIT, University of Toulouse
118, route de Narbonne
31042 Toulouse Cedex 9, France.
Racim.Fahssi@irit.fr

ABSTRACT

Task analysis can be considered as a fundamental component of user centered design methods as it provides a unique way of analyzing in a systematic way users' roles and activities. A widely used way of storing the information gathered during that phase in a structured and exhaustive way is to build task models which are then amenable to verification of properties or to performance evaluation. In widely used notations such as Hierarchical Task Analysis (HTA) or CTT (Concur Task Tree), information or objects manipulated by the users while performing the tasks does not receive a similar treatment as the sequencing of tasks which is usually carefully and exhaustively described. This paper proposes a systematic account for the various concepts manipulated by the users while performing tasks. Such concepts include different types of knowledge (declarative, situational, procedural and strategic), objects (manipulated by the user) and information. These concepts are systematically represented in a set of extensions of the HAMSTERS notation allowing the analysis of concepts-related properties such as learning curve, complexity, information workload, ... We demonstrate the application of the approach on the example of a two players game making explicit the connection between these extended task models and the user interface of the game.

Author Keywords

Task modelling; User Centered Design; Model-based approaches.

INTRODUCTION

To accomplish tasks, users may need to manipulate objects or information about current situation of the system and its environment, and knowledge about which actions to perform and

how to perform them. When designing interactive systems, the phase of tasks analysis usually focuses on: identifying goals which should be reached, grouping activities that have to be accomplished, understanding execution order of these activities and identifying objects required to perform the tasks. Existing analysis techniques and notations do not provide full support for explicitly and distinctively describing concepts related to the notions of Object, Knowledge and Information (OKI). In this paper, we focus on task modeling notations, which provide a widely used way of structuring and storing information gathered during the task analysis phase. Although some of them provide support for describing manipulated objects, most of these notations are focused on representation of procedures and methods for reaching a goal rather than the knowledge, information and objects involved. In this paper, we propose a systematic account for the various types of concepts required and manipulated by the users while performing activities including objects (manipulated by the user), information, and different types of knowledge (declarative, situational, procedural and strategic). We present how such information can be captured in task models provided some extensions are made. We exemplify these extensions on HAMSTERS notation. This proposal aims at providing the same level of support for describing concepts involved in users' tasks, as temporal ordering and hierarchical structuring of user tasks provided by most of the notations dedicated to tasks modeling. The underlying philosophy of the approach is similar to computer science view where, in the early days, data structure [9] was considered separately from control structure [9] until the unifying concept of object-oriented programming [9]. The example presented in the paper aims at demonstrating that integrating concepts and procedures within a single tasks model can provide multiple benefits throughout the development process from user needs identification to evaluation and training. The paper is structured as follows. Next section presents a review of literature from cognitive psychology research about mental representation of knowledge and allows us to identify various types of knowledge (such as declarative and procedural as well as situational and strategic). Section entitled "Knowledge, information and objects in task analysis and modeling" presents a review of the state of the art of task modeling notations to assess their capability of handling (in an explicit and systematic way) the types of knowledge identified in the previous section. Section entitled "Integrating knowledge into a procedural task modeling notation" summarizes the concepts in HAMSTERS notation for task modeling and proposes extensions to that notation by means of a concept map notation element (for knowledge, objects and information representation) and the integration of these concepts in the hierarchical description of tasks (with distinctive representations). These extensions are exemplified on a simple case study in section entitled "Illustrative example". Lessons learnt and conclusions are then presented in eponym sections.

KNOWLEDGE AND INFORMATION IN OUR MIND: STRUCTURE, MENTAL REPRESENTATION, AND LIFECYCLE

This section summarises contributions in the area of cognitive psychology dealing with the various types of knowledge processed by humans and how this knowledge is acquired, stored and retrieved. The Oxford Dictionary¹ defines knowledge as “facts, information, and skills acquired by a person through experience or education; the theoretical or practical understanding of a subject.” This definition exhibits two critical points. The former, “facts, information, and skills acquired by a person through experience or education” involves a relationship between facts, information and skills that have to be acquired to become knowledge. This acquisition can occur either through experience or through education. The latter, “the theoretical or practical understanding of a subject” suggests different types of understanding, theoretical and/or practical. These two aspects are discussed further in this section that is structured as follows. The first sub-section deals with types of knowledge. The second sub-section is dedicated to mental representation of each types of knowledge, and the third one describes knowledge lifecycle. The last one summarises the lessons learnt from previous sub-sections and identifies requirements to explicitly represent objects, knowledge and information in task models.

Types of knowledge

There are two main types of knowledge: declarative and procedural [9]. **Declarative knowledge** is factual knowledge that is true or false. It describes objects by specifying the properties which characterize them. This type of knowledge does not pay attention to the actions needed to obtain a result, but only to their properties [10]. Alternatively, **procedural knowledge** corresponds to the knowledge exercised in the performance a task. It is directly applied to a task and is acquired while performing activities [9]. Declarative knowledge corresponds to the “I know that” while procedural knowledge corresponds to “I know how”. They have different modes of mental representation and involve different memory processes and brain areas [10]. Beyond this declarative/procedural distinction, according to [9], there are other two types of knowledge: situational and strategic. Situational knowledge is related to case-based reasoning and contains domain specific information, while strategic knowledge is associated to a plan, a parallel checking and an analysis of possible choices. They have to be considered as refinements of the two main types (declarative and procedural).

The mental representation of declarative and procedural knowledge

After acquisition through senses, each sensed data is elaborated in different ways and using different cognitive processes to become information. Information can be elaborated as mental representations [9] which are hypothetical internal cognitive symbol that represents external reality, or a mental process that makes use of such a symbol [9].

Mental representation of declarative knowledge: structuring, grouping and connecting knowledge and information

According to [10], declarative knowledge is represented by mental images and/or symbolic forms such as words. The image is analogue to the object that it represents (e.g. a physical **object** or **information** in the real world) and is made up of concrete

attributes and features that can be concurrently captured. Conversely, the word is a symbolic representation and the relationship between this symbolic form and the object is arbitrary.

Knowledge structuring and grouping through concepts

A **concept** provides a mean for understanding the world and is a mental organization of knowledge. It is the fundamental unit of declarative knowledge, and often a single concept may be captured in a single word. They have different characteristics which ensure a wide flexibility in using them: such as (a) concepts can contain other concepts, (b) concepts can include typical and generic events, (c) concepts can present different abstraction degrees and (d) they can comprehend information about relationships [9].

Connecting through networks A semantic network is a network which represents semantic relations between concepts and can be represented as a web of interconnected elements made up of nodes and labels [9]. Nodes represent concepts and labels denote relationships which may involve category membership, attributes, or other semantic relations. These labels connect concepts in memory for allowing persons to link different nodes through their meaning. Semantic networks are given different names according to their scope and use. For instance, they are called associative networks when focusing on links and concept maps when focusing on concepts. As presented in [10] **concept maps** are also used as a direct method for assessing knowledge and can be used for representing the transition from novice to expert performance [9,10]. As claimed by Sowa [10] all these semantic networks share a graphical representation that can be used either to represent knowledge or to support automated systems for reasoning about the represented knowledge.

Mental representation of procedural knowledge: structuring, grouping and connecting knowledge and information

Mental representation of procedural knowledge has always been considered hard to identify and of course to prove and most of the results currently available derive from computer simulations. As claimed in [10] such knowledge could be represented as scripts which is a structure that describes sequences of events in a particular context. The structure is an interconnected net made of slots influencing each other.

Knowledge structuring and grouping through routines and subroutines

Procedural knowledge can be structured as a set of rules governing a production. As defined in [10] production system is the entire set of rules to perform a task or use a skill. These rules are organized into **routines** that are sets of instructions for accomplishing a task. They can be refined in **subroutines** that are sets of instructions to accomplish a subtask. These routines are typically organized according to basic control structures in computer science such as sequence, alternatives and iteration [9,9].

Connecting through hierarchical networks Due to the structuring in routines and subroutines it is possible to adopt a hierarchical representation to describe them [10]. In such hierarchical networks information known at a higher level in the hierarchy is known at lower levels.

Even though we presented declarative and procedural knowledge in an independent way, both knowledge types corresponds to different aspects of the same concept. For this reason, next section describes research contributions aiming at proposing integrative models for representing declarative and procedural knowledge.

¹http://oxforddictionaries.com/us/definition/american_english/knowledge#m_en_us1261368 last accessed February 2013

Mental representation of declarative and procedural knowledge adopting integrative models: structuring, grouping and connecting

Adaptive Character of Thought—Rational (ACT-R): structuring, grouping and connecting

This approach integrates a network representation for declarative knowledge and a production system representation for procedural knowledge. On one hand, declarative network includes storing and retrieving of information (called compilation process) and, on the other hand, procedural system implements the processes of automation or proceduralization [9]. According to ACT-R [9], all knowledge begins as declarative information. Procedural knowledge is learned by making inferences from already existing factual knowledge. New production rules are formed by the conjunction or disjunction of existing production rules [9]. This demonstrates the importance of dealing with declarative knowledge in an explicit and exhaustive way.

Parallel Distributed Processing (PDP): structuring, grouping and connecting

Neuropsychological research work [9] on amnesic patients and psychobiological studies on animals and human brain indicate that a large part of human cognition is based on **parallel processing** in which multiple operations are performed simultaneously. These experiments suggest that “non-declarative knowledge” may encompass a broader range of mental representations than just procedural knowledge [10]. All these forms of non-declarative knowledge are implicit and not easily explainable as they are tacit [9]. This highlights the needs for dealing in an explicit way with such information in order to be sure that concepts (and more precisely OKI) are carefully analysed and represented. The Parallel Distributed Processing (PDP) approach suggests that humans are able to carry out a large number of cognitive operations simultaneously by the use of countless numbers of neural processors located in the brain [10]. In neural networks the configurations, also called patterns, carry more information than single units generating **knowledge representation**.

Lessons learned from this related work and requirements for task modelling

This section has described research work in cognitive psychology around the notion of knowledge classification and structuring. This related work has identified two main types of knowledge that have to be taken into account when analysing users’ activities. The first one is **declarative knowledge** that can be structured and grouped using **concepts (objects or information** from the outside world and **knowledge** from the user), which can then be connected ending up in a semantic network or **concept map**. The second main type of knowledge is **procedural knowledge** which is structured by mean of **routines** and **subroutines**, which can be connected together and thus represented in a **hierarchical structure**. Two sub-types of knowledge should also be taken into account when analysing users’ activities: **situational** and **strategic**. Both sub-types help refining declarative and procedural knowledge to represent more precisely users’ activities. The integrative approaches suggest that declarative and procedural knowledge can be structured, grouped and connected together as they are **closely intertwined** while users perform tasks. This related work analysis allows us to identify a set of requirements for including in a systematic way both declarative and procedural knowledge not only during tasks analysis and also during tasks modelling:

- REQ_1: Concepts should be used to represent explicitly declarative knowledge.
- REQ_2: Concepts can be refined into objects and into information and these two entities should be made explicit.
- REQ_3: Semantic networks may be used to represent the concepts and their relationships.
- REQ_4: As concepts relate to each other, these connections should be made explicit to structure and group declarative knowledge.
- REQ_5: Declarative knowledge can be refined into strategic and situational and these two types of knowledge should be represented explicitly.
- REQ_6: Routines and subroutines should be used to represent explicitly procedural knowledge.
- REQ_7: As routines and subroutines are related to each other, these relationships should be made explicit in order to structure and group procedural knowledge.
- REQ_8: Hierarchical structures (trees) may be used to represent both routines and sub-routines and their relationships.
- REQ_9: Procedural knowledge can be refined into strategic and situational and these two types of knowledge should be represented explicitly.
- REQ_10: All the information above should be integrated in a single model as they are closely intertwined while users perform tasks.

It is important to note that this related work section has focussed on the data manipulated by users when performing tasks either being of knowledge, information or objects nature. Cognitive process manipulating this data (analysis and decision) or detailed interaction with the real world (motor and perception) have not been detailed in this related work as they are already rather extensively covered by notations for task models.

KNOWLEDGE, INFORMATION AND OBJECTS IN TASK ANALYSIS AND MODELLING

This section proposes a detailed analysis of a number of established task modelling techniques (including their notations and support tools) to assess their capability to cover the requirements detailed above. Some of the most recent contributions are Caffiau et al. [9], Villaren et al. [10] and Martinie et al. [9] demonstrating the still vivid interest from the scientific community for task modelling summarizes for each main type of knowledge (declarative and procedural, column 1), the requirements which have been expressed above (column 2).

Table 1. Comparison of task modelling notations according to suggested requirements (“√”=satisfied; “x”= not satisfied)

	Requirements /notations [reference]	HTA [9]	GOMS [9]	MAD [10]	TKS [9]	TAKD [9]	GTA [10]	UAN [9]	Diane+ [10]	VTML [9]	CTT [9]	CWA [10]	AMBOSS [9]	DSA [10]	HAMSTERS
Declarative Knowledge (DK)	REQ_1. Concepts used to represent DK	x	x	x	√	x	X	x	x	x	x	√	x	√	x
	REQ_2. Refinement of concepts into objects and into information	x	x	Partly with objects (explicit)	x	Partly with objects	X	x	x	x	Partly with objects	x	x	Partly with objects	Partly with objects
	REQ_3. Concepts connections should be made explicit	x	x	x	√	x	X	x	x	x	x	√	x	√	x
	REQ_4. Semantic networks to represent the concepts and their relationships	x	x	x	x	x	X	x	x	x	x	x	x	√	x
	REQ_5. Refinement of DK into strategic knowledge and into situational knowledge	x	x	x	x	x	X	x	x	x	x	x	x	x	x
Procedural Knowledge (PK)	REQ_6 and REQ_7. Routines, subroutines and their interrelationships to represent PK	x	x	x	x	x	X	x	x	x	x	x	x	x	√
	REQ_8. Hierarchical structures to represent routines, sub-routines and their interrelationships	√	√	√	x	x	√	√	√	√	√	x	√	√	√
	REQ_9. Refinement of PK into strategic knowledge and into situational knowledge	x	x	x	x	x	X	x	x	x	x	x	x	x	x
Integration	REQ_10. All the information above should be integrated in a single model	x	Through pre-condition	Through pre-condition	Through pre-condition	Through pre-condition	Through pre-condition	x	Through pre-condition	Through pre-condition	Through pre-condition	Through constraint	Through pre-condition	Through pre-condition	Through pre-condition
Tool Support		Task-Architect	CAT-HCI and others*	K-MAde	ADEPT	LUTAKD	GTA and EUTERPE	Quantum	TAMOT and Isolde	SpecTRM tool Suite	CTTe		AMBOSS	SAMANTA	HAMSTERS

As the tools associated to the notation sometimes cover only part of the notation and sometimes go beyond the notation (with the help of simulation or analysis for instance), last row to “Tool support” for each task modelling notation. If the box shows a tick “√” means that the task modelling notation satisfies the corresponding requirement, while if the box shows a cross “x” means that the task modelling notation does not satisfy the requirement. “Partially with objects” sentence indicates that the task modelling notation partially satisfies the requirements i.e. does not deal with knowledge and information. “Through precondition” or “Through constraint” sentences indicate that the task modelling notation has attempted to integrate all the requirements via precondition or constraint. Due to space constraints, only task modelling techniques are presented with references but the tools supporting editing and sometimes simulating models are indicated (without reference).

INTEGRATING KNOWLEDGE INTO A PROCEDURAL TASK MODELLING NOTATION

The comparison of existing task modelling notations has demonstrated the limited support for explicitly representing knowledge and for integrating it into the usual procedural-centred description of tasks. As a consequence, task models produced using these notations only capture partial information on the tasks which may in the end result in an increase the design effort (increased number of prototyping-evaluation cycles for example for correcting mistakes). In order to overcome this issue, our

proposal aims at providing support for representing in a complete and systematic way information and knowledge users need when accomplishing tasks while using interactive systems.

This section presents how a task modelling notation can be extended in order to provide support for complete and systematic description of required knowledge when using an interactive system. In this paper we used the HAMSTERS notation as a basis but this approach is also compatible with other procedural-centred notations such as CTT for instance. The approach is similar to previous work we have recently conducted in order to address scalability issues in task models by adding structuring mechanisms [9]. In this work HAMSTERS was also used to convey the concepts but the extensions are applicable to any other procedural-centred notation.

HAMSTERS task modelling notation and tool

HAMSTERS² is a tool-supported graphical task modelling notation aiming at representing human activities in a hierarchical and ordered way. Goals can be decomposed into sub-goals, which can in turn be decomposed into activities, and output of this decomposition is a graphical tree of nodes. Nodes can be tasks or temporal operators. Tasks can be of several types (as illustrated in Figure 1) and contain information such as a name, information details, critical level. Only the high-level task type are presented

²<http://www.irit.fr/recherches/ICS/software/hamsters/index.html>

here (due to space constraints) but they are further refined (for instance the cognitive tasks can be refined in Analysis and Decision tasks [9]).

For the procedural description concepts are very close to the ones of CTT such as the task types and the operators. Temporal operators are used to represent temporal relationships between sub-goals and between activities (as detailed in Table 2). Tasks can also be tagged by temporal properties to indicate whether or not they are iterative, optional or both [9].










Task type	Icons in HAMSTERS task model
Abstract task	 Abstract task
System task	 System task
User task	 User task  Perceptive task  Cognitive task  Motor task
Interactive task	 Interactive input task  Interactive output task  Interactive input output task

Figure 1. High-level task types in HAMSTERS

As explained above; composition and structuration mechanisms have been introduced in order to provide support for description of large amounts of activities [9]. One main element of these mechanisms is subroutine. A subroutine is a group of activities that a user performs several times possibly in different contexts and which might exhibit different types of information flows. A subroutine can be represented as a task model and a task model can use a subroutine to refer to a set of activities. This element of notation enables the distribution of large amount of tasks across different task models and factorization of the number of tasks. This corresponds precisely to the way procedural knowledge is mentally represented as explained in the dedicated section.

Table 2. Temporal ordering operators in HAMSTERS

Operator type	Symbol	Description
Enable	$T1 \gg T2$	T2 is executed after T1
Concurrent	$T1 \parallel T2$	T1 and T2 are executed at the same time
Choice	$T1 \sqcup T2$	T1 is executed OR T2 is executed
Disable	$T1 \triangleright T2$	Execution of T2 interrupts the execution of T1
Suspend-resume	$T1 \triangleright T2$	Execution of T2 interrupts the execution of T1, T1 execution is resumed after T2
Order Independent	$T1 \equiv T2$	T1 is executed then T2 OR T2 is executed then T1

HAMSTERS also provides support for representing how particular objects (data, information....) are related to particular tasks. Figure 2 illustrates the three relationships (input, output or both) between objects and tasks that can be expressed with HAMSTERS notation. Object (data, information...) can be needed as an input to accomplish a particular task (as illustrated in Figure 2a by the incoming arrow). Particular tasks may generate an object or modify it (as illustrated in Figure 2b)

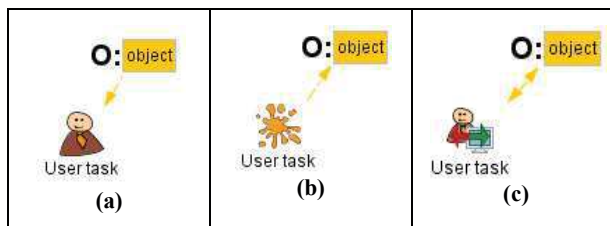


Figure 2. Relationships between tasks and objects in HAMSTERS

(such a precondition is graphically visible on the task at the top of the hierarchy of the task model in Figure 10) for the tasks and object flow between tasks can also be captured by means of input and output ports (as shown in Figure 3).

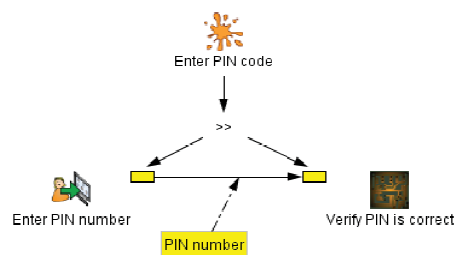


Figure 3. Input/output ports and their relationship with objects in HAMSTERS

Amongst the requirements identified in paragraph entitled “Lessons learned from this related work and requirements for task modelling”, HAMSTERS fulfils requirements 6, 7 and 8 thanks to its hierarchical decomposition and subroutine mechanisms, which can be used to describe procedural knowledge. HAMSTERS partially fulfils requirements 1 and 2 thanks to the explicit representation of objects that can be integrated in the task model by means of pre-conditions on tasks for instance (partly covering requirement 10).

Extensions to integrate knowledge representations

This section presents the extensions to HAMSTERS notation in order to allow the representation of all the elements of declarative knowledge and their integration within the procedural knowledge description of HAMSTERS. The main changes correspond to the explicit handling of procedural knowledge various elements within a concept map and the explicit representation of these concepts within the task tree.

Extensions for the representation of declarative knowledge

HAMSTERS capabilities to describe relationships between concepts and tasks have been extended. Knowledge of declarative type and its refinement in strategic and situational dimensions can be represented using the corresponding boxes illustrated in Figure 4. Relationships between the represented knowledge and the tasks can be represented using input/output relationships represented with arcs as for the objects.

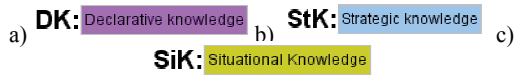


Figure 4. Representation of declarative knowledge a) which can be further refined into strategic b) and situational c)

Concepts which are of information type can also be represented in a non-ambiguous way using the Information box illustrated in Figure 5. As for an object, it is possible to represent the relationships between a task and information (input, output and input-output), as described in previous paragraph.

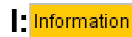


Figure 5. Representation of a concept of information type

Extensions for the representation of procedural knowledge
 HAMSTERS capabilities to describe relationships between tasks and temporal ordering operators have been extended. Representative distinctions between strategic and situational procedure can be made using two new types of arcs illustrated in Figure 6.

An ordered set of actions related to a strategy the user can apply will be highlighted with the blue “St” tagged arcs (Figure 6a). An ordered set of actions the user can execute in a given situation will be highlighted with the green “Si” tagged arcs (Figure 6b).

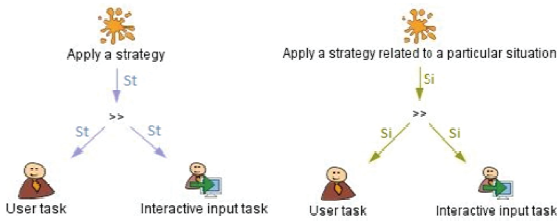


Figure 6. Representation of procedural knowledge refined into strategic a) and situational b)

ILLUSTRATIVE EXAMPLE

To illustrate how several types of knowledge can be represented in complete and systematic way during task modelling activities, and due to space constraints, we have chosen a simple example. Though this framework is used to analyse activities handled by users of interactive critical systems such as pilots and ground segments engineers, a case study would have required more space in order to describe the application domain concepts. Game of 15 is a two player traditional game. Each player takes his/her turn to choose a remaining number (a token) ranging from 1 to 9. The first player whose sum of selected number is exactly 15 wins the game. In this example, we analyse the tasks performed by the players using the interactive application depicted in Figure 7.



Figure 7. Example of a user interface for playing Game of 15

The upper part of the window display who is next player, below are displayed the remaining tokens while the last row displays the tokens taken by each player. More detailed information about behaviour of this application can be found in [9].

Figure 8 presents the concept map associated to the Game of 15. It represents required declarative knowledge necessary to be able to play the game. The concepts which are necessary or useful to perform actions during the game are grouped and linked through a semantic network. It is made up of 4 main nodes Objects, Roles, Rules and Strategies.

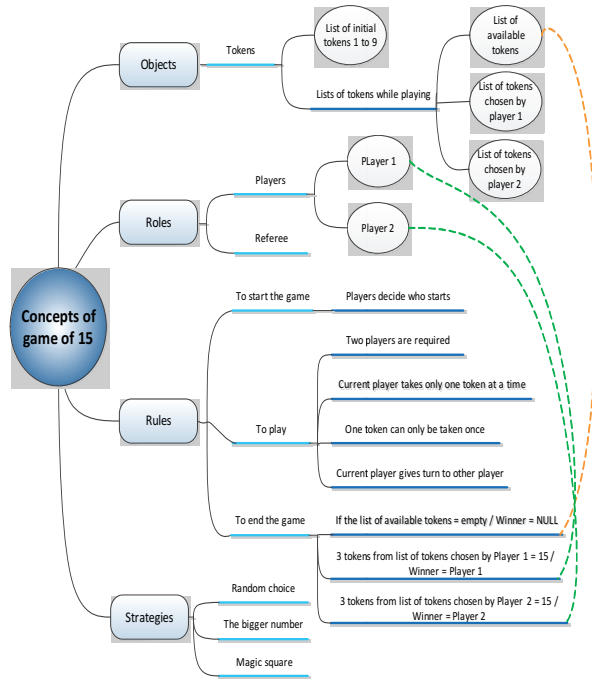


Figure 8. Concept map of game of 15

Structure from left to right shows the refinement of concepts (from abstract to concrete) as well as the instantiation (for instance from the concept of player to the instantiation of player1 and player2). Lastly, relationship between concepts is also made explicit (for instance the use of the link between the concept “list of available tokens” and the concept Rule). These concepts are integrated in task models depicted in Figure 9 and Figure 10 (presented top of next page) making explicit the integration of procedural and declarative knowledge in HAMSTERS.

Figures 9 and 10 present HAMSTERS models of tasks for playing Game of 15 using the interactive application presented in Figure 7.

Figure 9 presents the main HAMSTERS model with top-level task named “Play Game of 15”. The hierarchical and temporally

ordered sub-trees detail in a procedural way the various tasks that have to be known and performed by the players.

First they have to agree on who will start the game. Then they play the game in an iterative way (round-shaped arc at the top right of the “Play the game” abstract task) until they commonly agree to stop (if one player won or if there are no token left or if they just want to stop). The “Play the game” sub-task is a sequence of the following activities: turn taking (for one player), selecting a token (for current player, this task is a subroutine described in detail in Figure 10 and explained in next paragraph), processing combination of tokens (for each player), and giving turn to the other player.

In Figure 9, procedural knowledge required to play Game of 15 is described via hierarchical decomposition and temporally ordering of nodes. Declarative knowledge (refined in objects and rules in the concept map of Figure 8) is represented by mean of violet boxes connected to particular tasks with input arcs are they described required knowledge to accomplish a task. Required information is depicted by mean of orange boxes connected to particular tasks (with input arcs when information is mandatory to accomplish a task and with output arc when information is modified by a particular task).

Figure 10 describes in a hierarchical way the activities one player has to perform for selecting a token. In sequence, the interactive application interface is first displaying remaining tokens. The player then reads the set of available numbers, has to remember existing strategies and to adopt a particular strategy to select a

number (sub-trees detail possible choices of strategies and tasks which have to be executed according to the selected strategy). Once the player has selected a token, he/she inputs the number and the interface displays the token that has been selected. In this task model too, procedural knowledge required to accomplish “Select of token” subroutine is described via hierarchical decomposition and temporally ordering of nodes. Declarative knowledge (extracted from the concept map in Figure 8) is represented by mean of violet boxes connected to particular tasks with input arcs are they described required knowledge to accomplish a task. Strategic knowledge is represented both in a procedural way (using blue arcs in sub-trees dedicated to strategic knowledge below Adopt a Strategy task) and in a declarative way (blue boxes labeled StK (for **S**trategic **K**nowledge) connected to tasks related to particular strategies, also coming from the concept strategies in the concept map in Figure 8. Required information is depicted by mean of orange boxes connected to particular tasks. Example of situational knowledge description is not depicted in this example due to space constraints but would be represented in the same way as strategic knowledge (with corresponding color and arc label as described in Figure 6b)..

LESSONS LEARNT AND TOOL SUPPORT

This example demonstrates that integration of both declarative and procedural knowledge in a task model provides additional useful information for making explicit the concepts (objects, information and knowledge) required for reaching a goal as well as the concepts produced while performing the tasks.

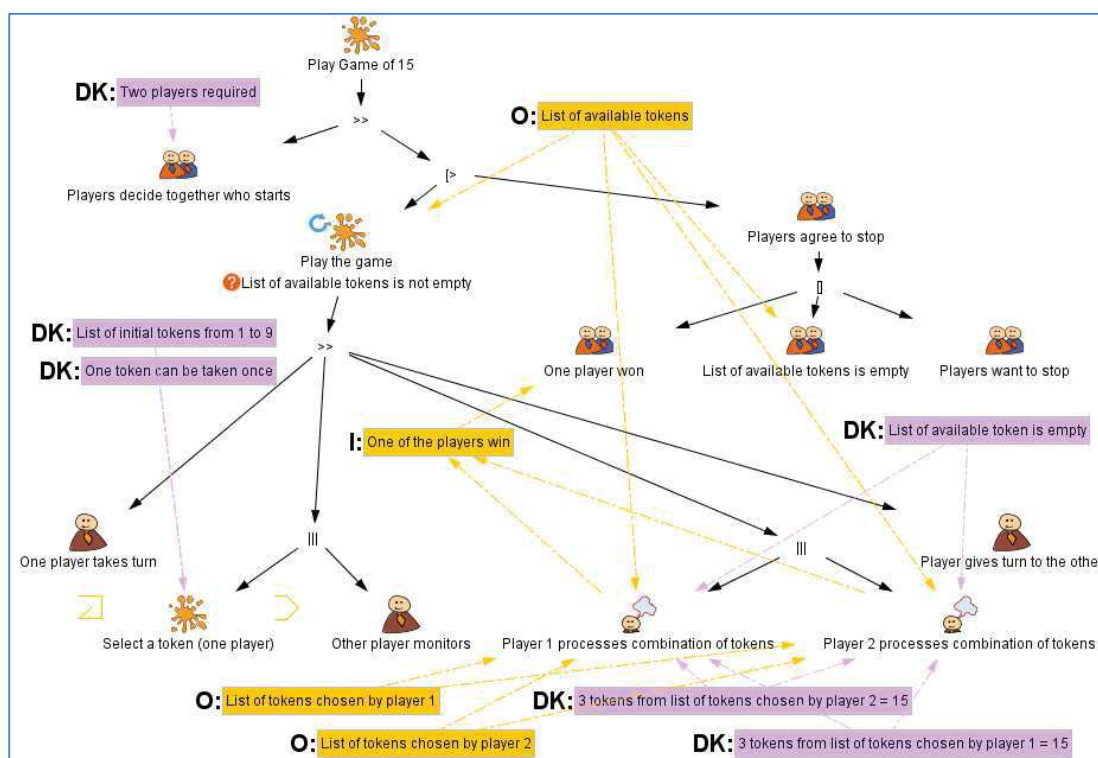


Figure 9. HAMSTERS model of game of 15

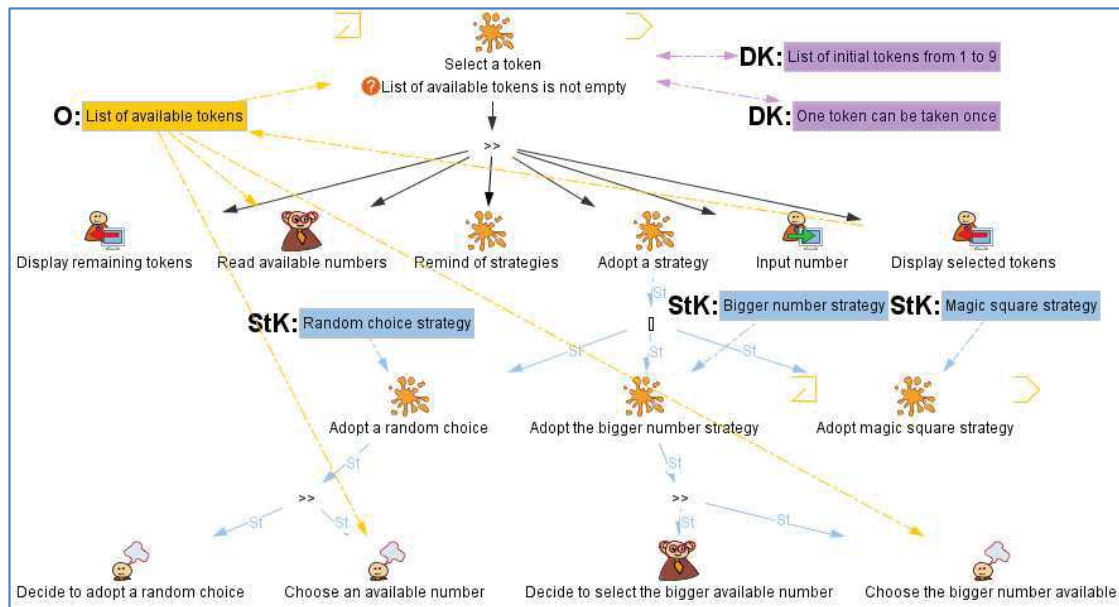


Figure 10. HAMSTERS model of game of 15-subroutine corresponding to the task Select token in Figure 9.

Such concepts can provide useful support in the various phases of the development process (provided that it explicitly refers to a task analysis and task modeling phases such as the one presented in [9]).

- **Requirements** by identifying what the users' needs in terms of knowledge, information and objects
- **Design** by making explicit the information that has to be provided to the user and the type of input that the users will need to perform
- **Implementation** by providing an informal but complete description of the data that the system will have to embed in order to allow performance of the user tasks
- **Evaluation** by providing an explicit representation of the information that will be manipulated and thus identifying candidate values for user testing.
- **Training** operators by making explicit what knowledge they have to master in order to use the system.

This systematic account for concepts used in tasks descriptions provides means for analysts to verify the compatibility and completeness of the concept map with respect to the task model. For instance one could check that all the concepts in the concept map appear in the task model and are graphically visible on the user interface. According to the case study this is the case for all the concepts around the notion of tokens such as list of available tokens, tokens selected by each player....

As mentioned above, this work has been done in the area of critical systems [9] where analysis about knowledge and information processing is of prime importance. For instance workload analysis has to be performed and decisions around tasks migration and automation can be fruitfully informed by a more complete account of declarative knowledge involved in tasks performance. All the extensions presented to the original

HAMSTERS notation have been implemented in HAMSTERS tool which allows both editing and simulating task models. In previous work [9] we have also demonstrated the possibility to connect task models with behavioural descriptions of user interfaces making co-execution and validation of models interactive. Extensions to the CIRCUS (Computer-aided-design of Interactive, Resilient, Critical & Usable Systems) tool suite (which integrates HAMSTERS and PetShop [9]) are currently under development for supporting the knowledge representation in a similar way.

CONCLUSIONS

This paper has presented how some declarative knowledge and procedural knowledge can be integrated in a notation for modelling tasks. We have grounded this contribution on a detailed study of related work both on cognitive psychology (where how human beings process and represent knowledge has been the focus of attention for many years) and on task modelling. Previous work in this field has mainly targeted at describing procedural knowledge leaving declarative knowledge as a second class citizen with the noticeable exception of [10] and [10] but the latter only consider procedural knowledge informally. We have shown on a simple game example that these two representations can be easily integrated providing additional benefits in terms of information flow in the tasks and in terms of information required for performing the tasks and information produced while performing the tasks. This work is currently used in the area of safety critical application (both for satellite ground segments and aircraft cockpits) with the double objective of supporting the early phases of the development process (requirements and design) as well as later ones (evaluation and training).

ACKNOWLEDGEMENTS

This work was partly sponsored by CNES (French National Space Studies Center) R&T Tortuga and MARACCASS as well as EUROCONTROL research network HALA! on Higher Automation Levels in Aviation.

REFERENCES

1. Anderson, J. R. Acquisition of cognitive skill. *Psychological Review*, 89, pp. 369-406 (1982).
2. Anderson, J. R. ACT: A simple theory of complex cognition. *American Psychologist*, 51, pp. 355-365(1996).
3. Anderson, J. R., John, B. E., Just, M. A., Carpenter, P. A., Kieras, D. E., & Meyer, D. E. Production system models of complex cognition. In Moore, J.D., Lehman, J.D (Eds.) *Proc. of the Seventeenth Annual Conference of the Cognitive Science Society*. Lawrence Erlbaum Associates, Hillsdale, NJ, pp. 9-12 (1995).
4. Barboni, E., Conversy, S., Navarre, D., & Palanque, P. Model-based engineering of widgets, user applications and servers compliant with ARINC 661 specification. In Doherty, G., Blandford, A. (Eds.) *Interactive Systems. Design, Specification, and Verification* (pp. 25-38). Springer Berlin Heidelberg (2007).
5. Barboni E., Ladry J-F., Navarre D., Palanque P., Winckler M: Beyond modelling: an integrated environment supporting co-execution of tasks and systems models. ACM SIGCHI conference Engineering Interactive Computing Systems (EICS 2010). Berlin, Germany, ACM SIGCHI, pp. 165-174 (2010).
6. Brown, M., Leveson, N. Modelling controller tasks for safety analysis. Int. Workshop on Human Error and System Development. Seattle, 1998. Available at <http://sunnyday.mit.edu/papers.html> (1998)
7. Caffiau, S., Scapin, D., Girard, P., Baron, M., Jambon, F. Increasing the expressive power of task analysis: Systematic comparison and empirical assessment of tool-supported task models. *Interacting with Computers*, 22, pp. 569-593 (2010)
8. Card, S. K, Moran, T. P, Newell, A. *The Psychology of Human-Computer Interaction*. Lawrence Erlbaum Associates, New Jersey (1983).
9. ACT-R Research Group website <http://act-r.psy.cmu.edu/> access date 01/23/2013
10. Codd E.F. Relational Database: A Practical Foundation for Productivity. *Commun. ACM* 25, 2, pp. 109-117 (1982).
11. Collins, A. M., Quillian, M. R. Retrieval time from semantic memory. *Journal of verbal learning and verbal behavior*, 8, 2, pp. 240-247 (1969)
12. de Jong, T., Ferguson-Hessler, M. G. M. Types and qualities of knowledge. *Educational Psychologist*, 31, pp. 105-113 (1996).
13. Diaper, D. Task analysis for knowledge descriptions (TAKD): A requiem for a method. *Behaviour and Information Technology*, 20, 3, pp. 199-212. (2001)
14. Giese, M., Mistrzik, T., Pfau, A., Szwillus, G., von Detten, M. AMBOSS: A Task Modelling Approach for Safety-Critical Systems. In Forbrig, P., Paternò, F. (Eds) HCSE/TAMODIA 2008, LNCS 5247, Pise, Italy, Springer, pp. 98-109 (2008).
15. Harel D. And/Or Programs: A New Approach to Structured Programming. *ACM Transaction on Programming Language Systems*, 2, 1, pp. 1-17 (1988).
16. Hix, D., Hartson, H. R. *Developing User Interfaces: ensuring usability through product and process*. John Wiley & Sons., New York, (1993).
17. Johnson, P., Johnson, H., Waddington, R., Shouls, A. Task related knowledge structures: Analysis, modelling and application. *People and Computers, IV*. Cambridge Univ. Press, pp. 119-144 (1989).
18. Koedinger, K. R., Corbett, A. Technology Bringing Learning Sciences to the Classroom. *The Cambridge Handbook of the Learning Sciences*. Cambridge University Press, pp. 61-78 (2006).
19. Komatsu, L. Recent Views of Conceptual Structure. *Psychological Bulletin*, 112, 4, pp. 500-526, (1992)
20. Marr, D. *Vision: A computational investigation into the human representation and processing of visual information*. MIT Press, London and Cambridge, MA (2010).
21. Martinie C., Palanque P., Navarre D., Barboni E. A Development Process for Usable Large Scale Interactive Critical Systems: Application to Satellite Ground Segments. In Winckler, M., Forbrig, P., Bernhaupt, R. (Eds.) *IFIP WG13.2 conf. on Human Centric Software Engineering (HCSE 2012)*, Springer, pp. 72-93 (2012).
22. Martinie, C., Palanque, P., Winckler, M. Structuring and Composition Mechanism to Address Scalability Issues in Task Models. In Campos, P.F., Graham, T. C. N, Jorge, J.A., Nunes, N.J., Palanque, P., Winckler, M. (Eds.) *IFIP TC13 Conference on Human-Computer Interaction (INTERACT 2011)*, Springer Verlag, LNCS 6948, 2011, pp. 589-609 (2011).
23. Navarre D., Palanque P., Ladry J-F., Barboni E. ICOs: A model-based user interface description technique dedicated to interactive systems addressing usability, reliability and scalability. *ACM Trans. Comput.-Hum. Interact.* 16(4) (2009)
24. Meyer B. *Object-Oriented Software Construction*. Prentice-Hall, isbn 0-13-629031-0 (1988)
25. Meyer, D. E, Annett, J., Duncan, K. D. Task analysis and training design. *Journal of Occupational Psychology*, 41 (1967).
26. Milner, B. Disorder of learning and memory after temporal lobe lesions in man. *Clinical Neurosurgery*, 19, pp. 421-446 (1972).
27. Navarre D., Palanque P., Winckler M. Task Models and System Models as A Bridge Between HCI and Software Engineering. In Seffah, A., Vanderdonckt, J., Desmarais, M.C. (Eds) *Human-Centered Software Engineering - Software Engineering Models, Patterns and Architectures for HCI. Human-Computer Interaction*. Series Springer, 357-385 (2009).
28. Newell, A. Production systems: Models of control structures. In: Chase, W. G. (ed.) *Visual information processing*. Academic Press, New York, (1973).
29. Paternò, F. *Model-Based Design and Evaluation of Interactive Application*. Berlin, Springer Verlag, (1999).
30. Pylyshyn, Z. W. Mental imagery: In search of a theory. *Behavioral and Brain Sciences*, 25, 2, pp. 157-238 (2002).
31. Rolls E. T. David Marr's vision: floreat computational neuroscience. *Brain* 134, pp. 913-916 (2011).
32. Royer, J. M., Cisero, C. A., Carlo, M. S. Techniques and procedures for assessing cognitive. *Review of Educational Research* Summer 63, 201-243 (1993).

33. Rumelhart, D. E., Hinton, G. E., McClelland, J. L. A general framework for parallel distributed processing. In: Rumelhart, D. E., McClelland, J. L. (eds.), *Parallel distributed processing: Explorations in the microstructure of cognition (vol. 2)*. MIT Press. Cambridge, MA, (1987).
34. Scapin, D., Pierret-Golbreich, C. Towards a method for task description: MAD. *Work with Display Units WWU'89*, Berlinguet, L., Berthelette, D. (Eds.) pp. 27-34 (1989).
35. Schank, R. C., Abelson, R. P. Knowledge and memory: The real story. In: Wyer, R. S. (ed.) *Advances in Social Cognition*. Lawrence Erlbaum Associates. Hillsdale, NJ, Vol.8, 1-86 (1995).
36. Shavelson, R. J., Ruiz-Primo, M. A., Wiley, E.W. Windows into the mind. *Higher Education* 49, 4, pp. 413-430 (2005).
37. Sowa, J. F.: Semantic networks. In: Shapiro, S. C. (ed.) *Encyclopedia of Artificial Intelligence*. John Wiley & Sons, New York, pp. 779-785 (1992).
38. Squire, L. R. Memory systems of the brain: A brief history and current perspective. *Neurobiology of Learning and Memory*, 82, 3, pp. 171-177 (2004).
39. Stanton, N.A., Stewart, R., Harris, D., et al. Distributed situation awareness in dynamic systems: theoretical development and application of an ergonomics methodology. *Ergonomics* 49, 12-13, pp. 1288-1311 (2006).
40. Sternberg, R. *Cognitive Psychology*. Holt, Rinehart and Winston, Inc. New York, NY, (1996).
41. Tarby, J., Barthet, M. The Diane+ method. In Vanderdonck, J (Ed). 2nd International Workshop on *Computer-Aided Design of User Interfaces*. Presses Universitaires de Namur, pp. 95-119 (1996).
42. Ten Berge, T., Van Hezewijk, R. Procedural and Declarative Knowledge: An Evolutionary Perspective. *Theory and Psychology* 9, 5, pp. 605-624 (1999).
43. Turban, E., Aronson, J. *Decision Support Systems and Intelligent Systems*. Prentice Hall, Inc. Upper Saddle River, NJ, (1988).
44. van der Veer, G. C., Lenting, V. F., Bergevoet, B. A. GTA: Groupware Task Analysis - modeling complexity. *Acta Psychologica*, 91, pp. 297-322 (1996).
45. Vicente, K. *Cognitive work analysis: Toward safe, productive, and healthy computer-based work*. Lawrence Erlbaum Associates, Mahwah, New Jersey, (1999).
46. Villaren T., Coppin G., and Leal A. Modeling task transitions to help designing for better situation awareness. In Diniz Junqueira Barbosa, S., Creissac Campos, J., Kazman, R., Palanque, P., Harrison, M.D., Reeves, S. (Eds.) *Proc. of the 4th ACM SIGCHI symposium on Engineering Interactive Computing Systems (EICS '12)*, ACM, New York, NY, USA, pp. 195-204 (2012)
47. Williams, C. G.: Using Concept Maps to Assess Conceptual Knowledge of Function. *Journal for Research in Mathematics Education* 29, 4, pp. 414-421 (1998).
48. Young, R. M. Production Systems in Cognitive Psychology. In: Smelser, N. J., Baltes, P. B. (eds.) *International Encyclopaedia of the Social and Behavioural*, Elsevier Science, Oxford, (2001).