Cooperative model generation for disabled people assistance

AISSAM BELABBAS, PASCAL BERRUET, ANDRE ROSSI, JEAN LUC PHILIPPE
LESTER, CNRS FRE 2734
Université de Bretagne Sud
Rue de Saint Maudé, Lorient, F-56321
FRANCE
http://web.univ-ubs.fr/lester/

Abstract: - This article presents a communication architecture of handicap technical assistance system in a multi users context. The proposed services are available in some particular areas only, for that, the environment is split into several areas and sub-areas, and services are associated to some area. Because many people live in this environment, we propose to assign for each area one manager and for each door in the living space one controller in order to guaranty availability of services (domotic services and navigation assistance). A dialogue protocol allowing the cooperation between users and their environment is defined and an enrichment sub-procedure is used to get a cooperation path model for each intelligent wheelchair.

Key-Words: Reconfiguration, Petri Nets, Cooperation, handicap assistance, manager.

1 Introduction

In the last few years, several associations and governments have developed a strategy within community development for the rehabilitation, equalization of opportunities, and social integration of people with disabilities in order to facilitate their access to appropriate health, education, social, vocational, and other services. (Agreed to in 1994 by ILO, UNESCO and WHO). The contribution of the scientific community is confirmed by the achievement of several projects [1], [2] [3] and implementation of a technical assistance for disabled people. Adaptability is the first feature of this kind of system. It enables some of the environment's equipment, such as kitchen counters, sinks, and doors, which can be added or customized to accommodate the needs of individuals with or without disabilities, or to accommodate the needs of persons with different types or degrees of disability (customized assistance). The architecture of technical assistance system is split into logical and physical parts [4]. The logical part describes services performed by the equipment of the system. The physical part describes the elements in the system (equipment) and the links between them. A service can be expressed as an ordered sequence of elementary services. Disabled people may need several services (mobility, domotic services, robotic services, communication services). The needs for services vary from one person to another because of the different impacts the handicap has on each individual's independence. In this paper, we propose a system designed to facilitate their everyday lives.

Nowadays, a system's ability to meet the intended Quality of Service and safety is a major issue, especially for a technical assistance system. As services are available at some specific places, it is necessary to improve reliability and to adopt an efficient reaction to failure in order to guaranty service availability. The reaction to failures can be considered on two different levels. The first one is called **reflex recovery** [5]. It's a local reaction because neither heavy communications, nor large pieces of information are required to make a decision (avoiding a collision for instance). In the proposed system, an obstacle avoidance system is supposed to be installed on any wheelchair of the house. The second level is called **reconfiguration** [6], [7], and is a global reaction. It essentially addresses a service availability objective. For example, in the context of handicap assistance, let us consider that watching television is only possible in the bedroom and in the living room. If the living room is no longer accessible to a particular user (because of a breakdown of the opening systems of some doors), the reconfiguration process consists in finding a new path to reach the bedroom. At the time of a reaction side, reconfiguration is carried out in progressive levels. This progression is performed according to the reaction complexity and to the time available for reconfiguration. The technical assistance system proposed in this paper is viewed as a reconfigurable system. The main reason for the growing interest in reconfigurable systems is the ability to respond quickly and efficiently to changes. The aim of providing assistance is to improve the navigation of an intelligent wheelchair within an adapted living space equipped with an environment control system. Managing the navigation of several users in an indoor environment made of automated doors and services requires cooperation between the intelligent wheelchairs and the environmental control system. Only discrete aspects of navigation are taken into account in this work. This paper proposes a method to generate the necessary coordination so as to allow for the cooperation between different elements of the considered system, in order to implement the corresponding transfers of a selected path [8]. The architecture of this system is presented in the next section. A brief recall of the basis of the proposed approach is given in section 3. The management of transfers is presented and modelled in sections 4 and 5. The principle of cooperation model generation and the use of the components are detailed in section 6. Section 7 presents prospective and concluding remarks. It has to be noticed that this paper deals with the dialogue protocols allowing for the cooperation between entities and the mechanism of generation of its corresponding cooperation model.

2 System architecture

The living space proposed in this paper is composed of two different entities: the intelligent wheelchairs and the environment control. Their cooperation enables them to deliver the requested services and to reconfigure the system if necessary. Fig. 1 depicts communication between them.

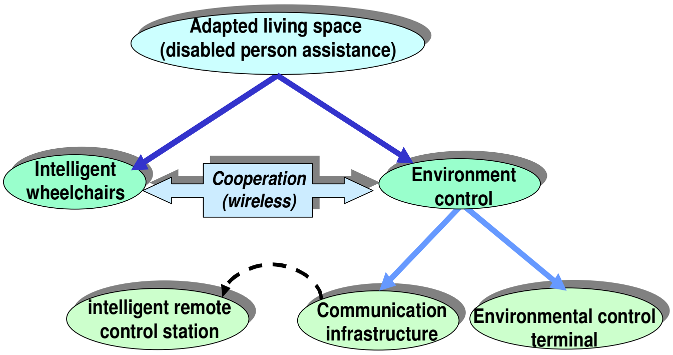
Adapted living space
(disabled person assistance)
Intelligent wheelchairs
Cooperation (wireless)
Environment control
intelligent remote control station
Communication infrastructure
Environmental control terminal

Fig. 1: The cooperating entities

2.1 The intelligent wheelchair

An intelligent wheelchair is an electric wheelchair equipped with an embedded system using Bluetooth™ technology, enabling it to communicate with the environment and to move alone. The communication interface enables it to cooperate with the environment entity. This cooperation is necessary to manage navigation (finding paths, going through automated doors), and to obtain access to areas. A Human-Machine Interface (customized joystick and PDA) is provided to let the user express needs for domotic services, and his or her assistance (the desired destination). Only discrete aspects of wheelchair navigation controls (moving forward/ backward, speeding up/slowing down) are taken into account in this work. The two following hypotheses are assumed.

- ***Hyp 1***: an obstacle avoidance system is supposed to be installed on any wheelchair of the house.
- ***Hyp 2***: the wheelchair can communicate its position at any time.
- ***Hyp 3***: the probability that the wheelchair falls in defect is nil.
- ***Hyp 4***: once a wheelchair receives an *authorization access* it starts the transfer immediately.

2.2 The environment control

The environment control is equipped with a communication infrastructure principally made of wireless connections using Bluetooth™ and WiFi technologies and an environmental control terminal to manage service house automations (automated doors, curtains, lights, and so on), as well as communication and the cooperation with intelligent wheelchairs. The system has to manage security (house accessibility), alarms (fire detection), and power saving policies. Most of the proposed services can be distributed into the corresponding control devices. Several automatic devices are equipped with Push-Buttons and Switches which serve as an interface for able people. The environment control terminal is equipped with a supervisor that has to guaranty the availability of the services even in the case of failure occurrence. Its reconfiguration strategy is mainly based on a path finding sub-procedure [8].

3 Proposed approach

Navigation in the context of handicap assistance is a major service. As a matter of fact, in the context of technical assistance for disabled people, an important characteristic is that services are strongly linked to areas. Thus, in the rest of this paper, a service is considered as available if and only if the user is able to reach at least one specific area of the considered living space. For example, if a user needs a service, he or she has to move to the area where this service is available. Thus, the problem of service availability is only a problem of area reachability. The proposed approach is divided into two different procedures. The first one is an off-line procedure, as illustrated in Fig. 2 (a). Actually, the environment must deal simultaneously with the request of all available users (wheelchairs, able people). At this stage, we propose doing so by assigning one manager to each area and one controller to each door. The behavior of the area manager and door controller is detailed afterward. The off-line procedure is run once. The second procedure is an on-line procedure, as illustrated in Fig. 2 (b), and is a sequence of three sub-procedures.

3.1 The path finding sub-procedure

Reconfiguration requires a model that supports the mechanisms of the reconfiguration strategy, and has to be able to express all the potentialities (services to be delivered) of the studied system. Thus, it is necessary to break up the adapted living area into areas and sub-areas, and associate the corresponding services to each area. This decomposition enables the management of the dependence between accessibility and environment control, and to obtain a model called "Topological Model" of the adapted living space. It is a non-deterministic model of all the possible paths from any area to any other one. The purpose of the topological model exploration, using a Path finding sub-procedure, is to deduce a deterministic model (i.e. a unique path from the current position of the user to the destination area), that is called the 'Topological Path model' for the considered wheelchair. After each failure occurrence, the topological model is updated. A new path can be computed if necessary. A complete description of this step can be found in [8].

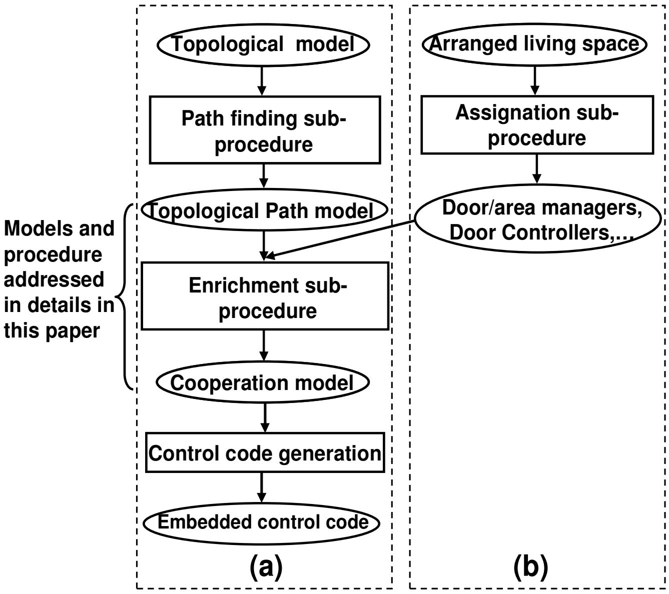

Fig. 2: A synopsis of the considered approach

3.2 The enrichment sub-procedure

This sub-procedure uses the topological path model as main input in order to return another deterministic model, called the cooperation model which is considered in this paper. More details about this model are presented in the next sections. The aim of this sub-procedure is to enrich, with patterns form library, the topological path model with all the necessary dialogue mechanisms between the wheelchair and its domotic environment (through the assigned managers and controllers) in order to obtain the corresponding cooperation model.

3.3 The control code generation

The cooperation model is itself the input of the control code generation sub-procedure that returns the embedded control code of the wheelchair. The whole procedure is run every time a new path is computed.

4 Management of transfers

4.1 Introduction

The path finding sub-procedure returns a valid path for every user. Each path is represented by a sequence of elementary transfers. An elementary transfer is a unidirectional move connecting two adjacent sub-areas (A and B) [8] (see Fig. 3). An intermediate sub-area (AB) is introduced between (A) and (B) in order to represent the position of the wheelchair during the transfer. As many users can move simultaneously in the adapted living space. Thus, access conflicts for a given sub-area may occur. This is the reason why a manager is associated with each sub-area. This manager has to ensure the respect of all constraints specifications (area capacity and access priority) related to its associated area. As can be seen in Fig. 3, two managers have been defined: manager of (A) and manager of (B).

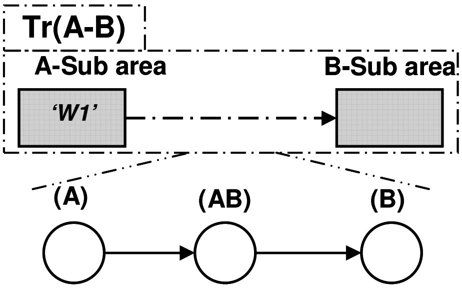

Fig. 3: An elementary transfer between (A) and (B)

Let's now define the manager's role. The first constraint is the capacity of sub-areas. In this study the sub-area capacity is supposed to be set to one. In order to solve the conflict access problem to the sub-areas, a priority access strategy is implemented. At this stage of the work, fixed priorities are considered. The highest priority is assigned to the user that needs the highest level of assistance *(fully automatic navigation)*. For the verification of the access priority constraint, any user has to be assigned a Priority Label (PL). These labels must be updated when the Access Priority Level (APL) of users changes. The three different types of transfers path may take part in a path are detailed in the next section.

4.2 Types of transfers

4.2.1 Unidirectional transfer

An unidirectional transfer is an elementary transfer (defined in section 4.1). Two roles are introduced to perform this transfer:

- ***Rule 1***: one wheelchair can reach destination area of transfer only if the manager of this area delivers an access authorization to the wheelchair.
- ***Rule 2***: once the wheelchair is in destination area, its manager releases the start area.

The mechanism of the unidirectional transfer is illustrated in Fig. 4. Simple arcs represent transfers between sub-areas. A transfer can be performed only if its associated condition (*if reservation of a sub area(..)*) is satisfied.

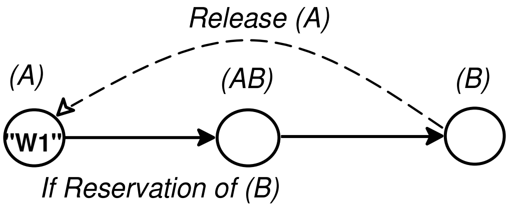

Fig. 4: Mechanism of an unidirectional transfer between (A) and (B).

The stippled arc represents the release of area (A). Let's consider a wheelchair denoted W1 in (A). After sending request access to area (B), W1 can reach (B) only if the manager of (B) delivers an *access authorization* to it (***Rule 1***). That is to say, wheelchair W1 starts the transfer toward (B) only if (B) has been reserved to W1 by its manager. Once W1 is in (B), the manager of (B) releases (A), allowing another wheelchair to enter (A) (***Rule 2***). Once the transfer is started (W1 in area (AB)), rule 2 allows W1 to go back in (A) if (B) is no longer reachable. These rules of conduct are necessary to manage simultaneous transfers, but they are not sufficient because blocking may happen. For example, let's consider another wheelchair W2 in (B). W2 (respectively W1) cannot reach (A) (respectively (B)) because it is engaged by wheelchair W1 (respectively W2). In order to avoid this case, the crossing of two wheelchairs must be possible. This situation is presented below.

4.2.2 Bi-directional transfer

A bi-directional transfer can be viewed as a set of two unidirectional transfers connecting two adjacent sub-areas (A and B). Carrying out both transfers simultaneously allows for the crossing of two wheelchairs. In this case, the two unidirectional transfers form a loop. The rules of conduct defined in the case of unidirectional transfer are not sufficient to perform the two transfers of a loop. In this particular case, the following rule of conduct has to be enforced:

- ***Rule 3***: access authorization must be delivered to the wheelchairs when the corresponding bidirectional transfer occurs.

This rule allows for the two unidirectional transfers to be performed simultaneously (crossing of two wheelchairs).

To perform the crossing of the wheelchairs, a loop manager *(ab:Manager)* is necessary. It is settled at a higher hierarchical level than the managers of (A) and (B) (a*:Manager, b:Manager*.). Its mission is to assist the two sub-areas managers of a loop at the time of a Bidirectional transfer. It should be noticed that to carry out any transfer, wheelchairs communicate only with their destination sub-area manager. However, at the time of a bidirectional transfer, the two managers will not be able to deliver the access authorizations. In this case, the loop manager intervenes in order to deliver the access authorizations of the two wheelchairs to their corresponding managers (***Rule 3***). To illustrate this mechanism, a UML compatible sequence diagram, is given in Fig. 5 [9],[10]. This figure shows the dialogues that take place when two wheelchairs have to pass each other. W1 Wheelchair wants to reach (B) from (A).

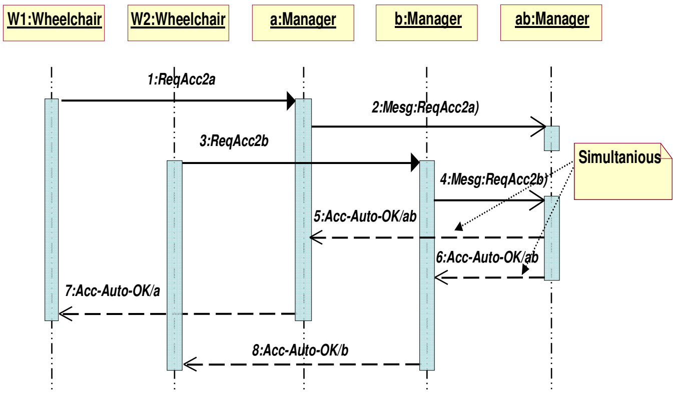

Fig. 5: Sequence diagram for the bi-directional transfer

According to the unidirectional transfer protocol, it sends a request *(1: ReqAcc-2a)* to the *a:Manager* in order to receive an access authorization. The *a:Manager* cannot deliver the corresponding access authorization because another wheelchair W2 fills the (A) area. The *a:Manager* sends a message *(2:Mesg(ReqAcc-2a))* to the *ab:Manager* (loop manager). Symmetrically, W2 wants to reach (A) from (B). Thus, it sends a request *(3: ReqAcc-2b)* to the *b:Manager*. In this case, b*:Manager* does not authorize these transfers, because they only meet the first rule of conduct (unidirectional transfer). Immediately after reception of request *(3: ReqAcc-2b)*, the b*:Manager* immediately sends a message *(4:Mesg(ReqAcc-2a))* to the ab*:Manager*. In this case, ab*:Manager* delivers the access authorizations for area managers in order to perform the bi-directional transfer. To do so, it simultaneously sends *(5: Acc-Auto-OK//ab*) and *(6: Acc-Auto-OK//ab*) messages respectively to the two managers *a:Manager* and *b:Manager*. Each manager sends the corresponding authorization to the wheelchairs (*a:Manager,* respectively *b:Manager, sends (Acc-Auto-OK//a)* respectively *(Acc-Auto-OK//b)* to *w1:Wheelchair,* respectively *w2:Wheelchair)*. Once both wheelchairs have received their access authorization, they begin their transfer. Violation of capacity constraints is impossible because of (***Hyp 4***). This rule is enforced in the cooperation model that is detailed in the section 5.

4.2.3 Going through a door

The third type of transfer is going through a door. The door area is split into two sub-areas (B and C) in order to represent both sides of a door because it's an access point between two areas (see Fig. 6). Two different steps are considered to carry out this transfer. The first step is the reaching of the door area. It's considered as an ordinary sub-area, thus an ordinary area manager is assigned to each of them. Sub-areas of the door have the capacity to accept only one wheelchair at a time (*capacity <B> + capacity <C> = 1*). This constraint ensures that no more than one user is going through the door at any time. Thus, the rules of conduct defined in unidirectional and bi-directional transfers are sufficient to manage the access to the door area.

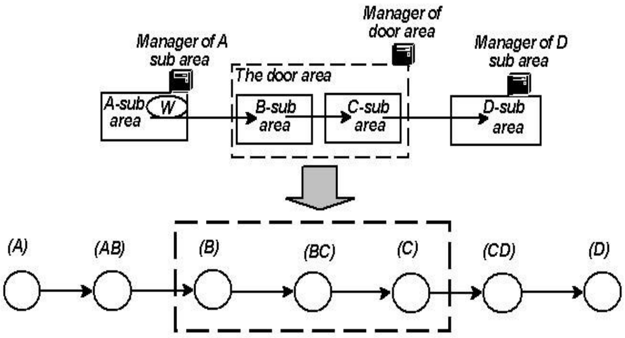

Fig. 6: The transfers of passing through a door.

The second step of transfer is the going through the door itself. The wheelchair will have to reach the other sub-area by passing through the door, it should be checked that the door is open before starting the transfer (passing through a door). To do so, the door controller delivers the access authorization after opening the door. To go throuth a door, special speed is used by wheelchair.

A communication architecture is illustrated by the collaboration diagram repsentend in Fig. 7. It's a UML diagram [11]used to illustrate object interactions in graphs or networks. A link line is used to represent a connection path between two objects. A sequence number is added to show the sequential order of messages in the current thread of control. For example, to perform the first step (reaching a door area), the wheelchair communicates with the door controller using a sequence of request/achnowledgement. It sends a request (*1.1:Request)* to access the door area, and it receives the access authorization under the form of an Achnowledgement (*1.2:Achnowledgement).*

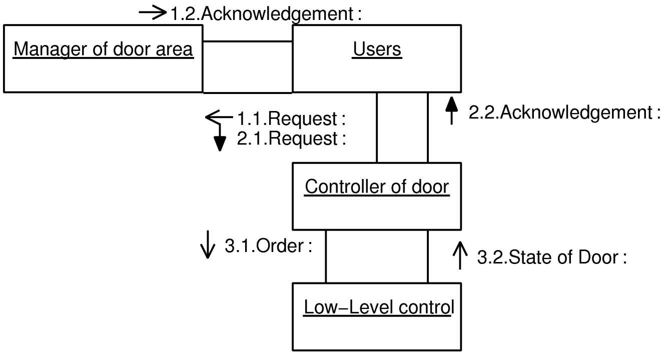

Fig. 7: The door collaboration diagram

In this considered architecture, each door is equipped with a low-level controller. Its mission is to perform the orders of opening and closing the door, and to broadcast the door state at all times. A door controller considered as an inteligent interface is placed on the higher level. It is necessary to filter the requests *(2.1:Request-Open/close the door, 2.1:Request (ReqAcc))* which are addressed to the door, as several persons may evolve in the environment. For example, if the door is open *(3.2:State of Door: Open),* the controller may not send another high level control command *(3.1:Order:Open)* to open it. The closing of the door is carried out according to the active operating mode. Several operating modes can be defined for each piece of equipment in the environment [12]. Only two states for the door operating mode are considered: fully automatic and semi-automatic states. In the fully automatic state, the door controller closes the door after the user goes through it, but in the semi-automatic state, the door is closed after the request of a user. The controller is informed by mode selector switch about the current active mode and selected state. This mechanism is not addressed in this paper. The state of active operating mode information is necessary to engage the closing of the door. For example, in the fully automatic operating mode, after the door has been gone through, the controller waits for a short time and automatically engages the closing of the door if it does not receive any request to open the door. Actually, in the cooperation process to go through the door (second step), the controller makes sure that the door is opened (using 3;2: S*tate of Door* message, see Fig. 7) and delivers the crossing access authorization to the wheelchair to go throuth in order to reach the other side of the door. The global process require wheelchair and environment cooperation, the generation of such a model is detailed in the next section.

5 Mechanism of cooperation

5.1 Dialogue protocol

Each wheelchair is assigned a path. The corresponding topological path model must be enriched with all the necessary dialogue mechanisms in order to allow for the cooperation between the wheelchair entity and the environment. This final model is the cooperation model. The following dialogue protocol is proposed to implement a cooperation process. Let us consider a wheelchair that has to carry out an elementary transfer. It first has to send a request to the environment entity. This request is processed by the manager of the destination sub-area that delivers the access authorization to the wheelchair. Once the wheelchair reaches its destination sub-area, it informs the environment entity by sending a message. The environment entity confirms the message reception by sending an acknowledgement. Thus, the dialogue protocol is based on a sequence of requests from the wheelchair entity and the corresponding acknowledgments from the environment entity. An illustrative cooperation model can be seen in the next section (See Fig. 12).

5.2 Modeling tools

A natural and easy way to design the cooperation model is to use Petri Nets [13], [14]. This modeling tool has many advantages. The main one is that a Petri Nets enables one to model communicating processes. A Petri nets allow for monitoring of the state of shared resources (like sub-areas), and manage them efficiently. They appear to be the most appropriate tool to model such capacity constraints. Petri Nets are not restricted to design modeling. They provide help with the analysis of the modeled system. Several approaches have been developed for qualitative and quantitative analyses of Petri Net models. Time Petri Nets (TPN) can be used to model concurrent, asynchronous and non-deterministic systems (model a priority), see Fig. 8. This work uses TPN formalism to check that the model behaves as expressed. An analyzer called Romeo [15], is run for that purpose. Using a design helper, Romeo implements an on-line simulation (TPNs, Scheduling-TPNs) a reachability model-checking (TPNs). It enables the early detection of some modeling errors during the design phase. Properties over markings can be expressed and tested. It is thus possible to test the reachability of marking such that it verifies *$(M(P_1) = 1 \vee M(P_3) \geq 3)$* where M ($P_i$) is the number of tokens in the place P_i of the net. The Time Petri net model of the manager of an ordinary sub-area (A) is shown in Fig. 8. The communication ports of the manager with the cooperation path model are represented as places with double-rings. The capacity of each area is represented by a colored place (*Cap(A)*). To release start transfer area (Z_i) (***Rule 2***), T5 (respectively T6) is fired and one token enters place *Cap (Z_j)*. To simplify the illustration, two access priority levels (High/Low Access Priority Level: HAPL/LAPL) are represented in the PN manager model. This priority is enforced by time transitions T3 and T4. For each path, its corresponding path model has been built using Petri nets. The Petri net model of a door controller expresses the behavior of the door controller. A low-level control model of a door is built using Petri nets. It has five communication ports with the door controller (input ports: orders; output ports: states of the door). The other places model the three states of the door motor.

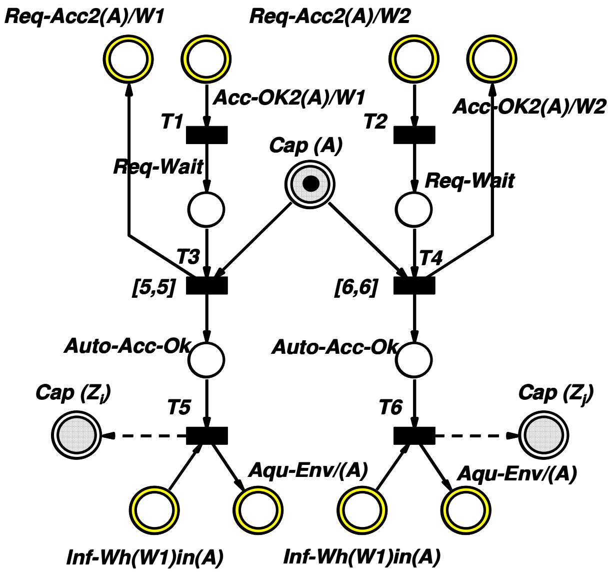

Fig. 8 : Petri net model of the sub-area manager

6 Cooperation model

The topological path model is itself the input of the enrichment sub-procedure. It's important to notice that there are as many Topological Path models as there are wheelchair users in the living space. For the sake of simplicity, only one user is represented. Let's consider the computed path represented in Fig. 9**.**

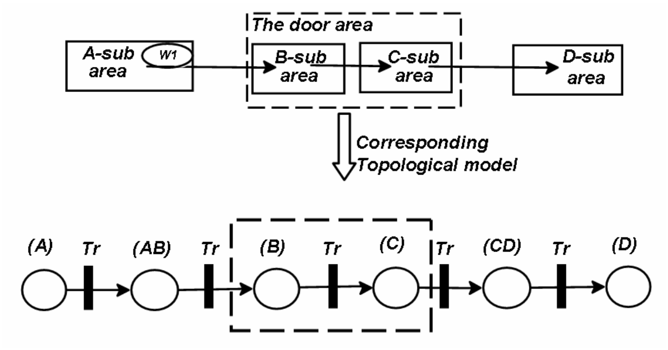

Fig. 9: Example of a topological path model

The Petri net model of path ABCD is given in Fig. 9. Sub-areas are represented by places, and the transfers between them are represented by transitions. This paper identifies two fundamental patterns that apply to communicating processes. Thereafter, a modular design using this pattern is used to build the cooperation path model. A pattern is viewed as a sequence of requests/acknowledgement modeled with Petri nets. They are two types of patterns in library. The first one is called the access authorization pattern. It's necessary to reach the adjacent sub-area. Its corresponding Petri net model is depicted in Fig. 10. It can be easily instantiated. For example, Req-Acc2(B)/W1 is the access request made by wheelchair W1 from the manager of (B). The Acc-Ok2(B)/W1 is its corresponding access authorization. The place named Wait-In(A) models W1 waiting for the acknowledgement in (A).

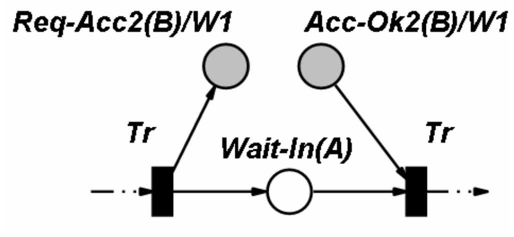

Fig. 10: Petri net model for the access authorization pattern

The second pattern expresses the acknowledgement when the destination is reached (See Fig. 11). It's called the informational pattern. For example, *Inf-Wh(W1)in(B)* is the information request made by the W1 wheelchair to the manager of the (B) sub-area in order to inform it that it has arrived. The *B:manager* confirms the information by the *Aqu-Env/(B)* acknowledgement. Once the patterns are defined and added to the library elements, it can henceforth be used by the enrichment sub-procedure to obtain the cooperation Path model. In this step, the parameterization of patterns allows for the specification of the fusion of communication places with the other Petri nets models (managers, controller,…). This mechanism is illustrated in the Fig. 12.

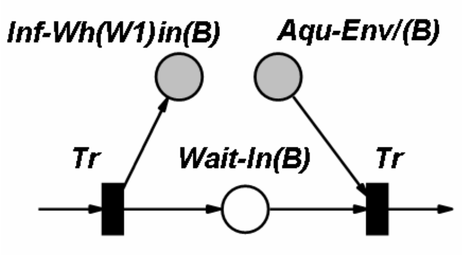

Fig. 11: Petri net model of the informational pattern.

Its principle is to substitute the transfer transitions between the sub-areas with the corresponding communication patterns as is shown in Fig. 12**.**

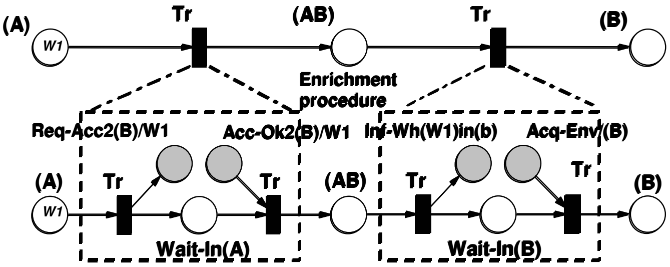

Fig. 12: Example of enrichment sub-procedure

Up to here, we have generated the different models that are necessary to follow the computed path. Romeo is used to check that the model behaves as expressed. But the fusion of the communication places has to be performed to implement the cooperation between both entities (view the example in Fig. 13).

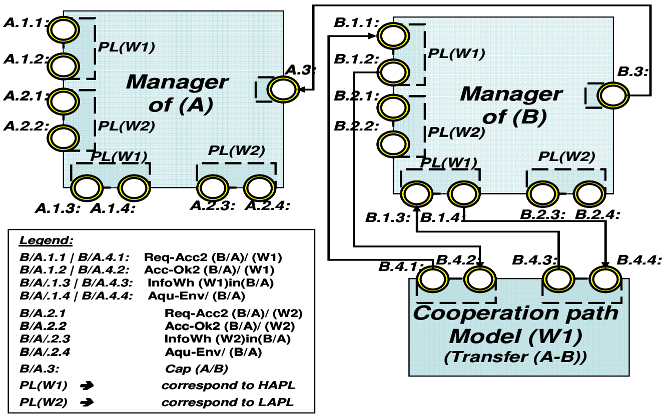
A.1.1:
A.1.2:
A.2.1:
A.2.2:
PL(W1)
PL(W2)
Manager of (A)
A.3:
PL(W1)
PL(W2)
A.1.3:
A.1.4:
A.2.3:
A.2.4:
B.1.1:
B.1.2:
B.2.1:
B.2.2:
PL(W1)
PL(W2)
Manager of (B)
B.3:
PL(W1)
PL(W2)
B.1.3:
B.1.4:
B.2.3:
B.2.4:
B.4.1:
B.4.2:
B.4.3:
B.4.4:
Cooperation path Model (W1)
(Transfer (A-B))
Legend:
B/A.1.1 | B/A.4.1: Req-Acc2 (B/A)/ (W1)
B/A.1.2 | B/A.4.2: Acc-Ok2 (B/A)/ (W1)
B/A/.1.3 | B/A.4.3: InfoWh (W1)in(B/A)
B/A/.1.4 | B/A.4.4: Aqu-Env/ (B/A)
B/A.2.1 Req-Acc2 (B/A)/ (W2)
B/A.2.2 Acc-Ok2 (B/A)/ (W2)
B/A/.2.3 InfoWh (W2)in(B/A)
B/A/.2.4 Aqu-Env/ (B/A)
B/A.3: Cap (A/B)
PL(W1) ➔ correspond to HAPL
PL(W2) ➔ correspond to LAPL

Fig. 13: Example of communication between the different elements

7 Conclusion

This paper presents an enhanced sub-procedure used to transform a topological path model into a cooperation path model. This sub-procedure uses two generic patterns applied to communicating processes to build the cooperation path model. In order to simultaneously manage the movement of all users that evolve in the environment, drive rules are defined. A manager ensures that these rules are respected. The cooperation path model is modeled using Petri net which is the most appropriate tool to model such capacity constraints and communicating processes. A check of Petri net classical properties (boundedness, deadlocks, liveness) for models and the expected behavior using tool Romeo was carried out. Using Petri Nets allows to validate the required behavior of PN models (manager/controller, cooperation path model) during the conception stage, and to validate the enrichment sub-procedure. The next step of this work is to generate a safe navigation source code (control code generation step) corresponding to each cooperation path model and its implementation.

References:

[1] Colle. E ; Rybarczyk, Y ; Hoppenot, P ; *« ARPH :an assistant robot for disabled people »* SMC 2002, Hammamet, Tunisie 6-9 October 2002Belabbas, A.

[2] Dario, P ; Guglielmelli, E ; Laschi, C ; Teti, G ; *« MOVAID : a personnal robot in everyday life of disabled and elderly people »* Technology and disability journal, n°10, IOS Press, Pays-Bas , 1999, p. 77-93.

[3] Leloup, J; Gaucher, P; "*The HMPH project: software for the design of an adapted living area*" 8th International conference on computers helping people with special needs ICCHP'2002, Autriche, 15-20 juillet 2002.

[4] Frizon de Lamotte. F, Berruet.P and Philippe. J.L « *A model for the reconfiguration of manufacturing systems* ». IFAC World congress, Prague, July 2005.

[5] Berruet, P. Coudert, T. Philippe, J, . *"Integration of dependability aspects in transitic systems",* IEEE-IMACS CESA 2003, Lille, 2003.

[6] Toguyeni, A.K.A ; Berruet, P. and Craye, E. "*Models and Algorithms for Failure Diagnosis and Recovery in FMS*", International Journal of Manufacturing Systems, Kluwer Academic Publisher Vol. 15, Janury 2003, pp. 57-85.

[7] Gehin, A, L ; Assas, M. and Staroswieki, M. "*Structural Analysis of System reconfigurability*", IFAC/SAFEPROCESS' 2000 Symp. on "*Fault Detection, Supervision and Safety for Technical Processes*", Budapest (Hongrie), Vol.1, 2000, pp. 292-297.

[8] Belabbas, A. Berruet, P. Rossi, A. Philippe J-L.: "*A reconfigurable technical assistance for disabled people*". 17th IMACS World Congress. Paris, France 11-15 july 2005. Paper Number: T4-R-00-0627.

[9] Booch. G, Rumbaugh. J, Jacobson. I *« UML users guide »* addision-wesley, 1999.

[10] Dumas, M., ter Hofstede, A.H.M: *UML Activity Diagrams as a Workflow Specification Language*, UML, LNCS 2185 , 2001, pp. 76–90.

[11] Larman. G "*Applying UML and Patterns*" Prentice Hall edition, 2004.

[12] Dangoumau ,N. Craye, *E* . MRS '01. Modélisation des systèmes réactifs . « *Commutation de modes dans un système de production* », Toulouse (France), 17-19 octobre 2001 pp 377-392.

[13] Murata, T. "*Petri nets: properties, Analysis and Applications*" IEEE Computer, Vol 77, No 4, April 1989. Pp 541-580.

[14] Li, X, and Lilius, J, "*Verifying Time Petri Nets by linear Programming*", Technical Report 259, Turku Centre for Computer Science TUCS, 1999.

[15] Gardey, G; Mullins, J and Roux, O (H)."*Non-interference control synthesis for security timed automata*". *In* 3rd International Workshop on Security Issues in Concurrency (SecCo'05)*, Electronic Notes in Theoretical Computer Science, San Francisco, USA, August* Elsevier *2005.*