

# Back on Track: Lost and Found on Public Transportation

*Stefan Carmien and Michael Obach*

<stefan.carmien, michael.obach> @tecnalia.com  
TECNALIA Research & Innovation  
Paseo Mikeletegi, 1  
20009 Donostia - San Sebastian, Spain  
+34 946 430 850

Errors, in particular human errors, play an important role in many aspects of human life, from day-to-day activities to extraordinary situations. This paper describes the theoretical background in the context of Distributed Cognition and the practical design process of an error trapping and mitigation system for supporting seniors' (including disabled seniors) use of public transportation.

Data available from the ASSISTANT project support the basic assumption that there are not many things that typically go wrong when people from this target group use buses, trams, trains etc., but that these classes of errors cover a majority of instances. An error model, being a first approximation of a rule-based error capturing and mitigation system, is proposed that is adequate for sparse data and available before the initial use of the system.

Several error types, sources of information coming from a Personal Navigation Device and reasonable conclusions are presented and discussed. Furthermore, some examples of an error trapping and mitigation class tree are provided, as well as some aspects of implementing these systems in earlier projects.

The on-going ASSISTANT project addresses especially mitigation types and error type classification, which can lead to easier implementation and broader acceptance in a near future.

## Introduction

Things break, things get lost, you get lost – these are events that happen all the time. The Ambient Assisted Living (AAL) project ASSISTANT, supporting elders' use of public transportation, acknowledges these facts and has an active system of support to capture and mitigate these errors. This paper describes the design process of an error trapping and mitigation system based on research into human error [8, 15, 16] and working with representative end-users. ASSISTANT will capture errors and group them into categories and select and deliver, on the basis of a user model, the appropriate mitigation strategies.

## State of the Art and Previous Work

Considering the role that human error plays in both day-to-day life and extraordinary situations, there has been a relative paucity of work done in this field. Early research was done by Don Norman [14] in his work on slips and mistakes; James Reason [15] took this taxonomy further by discussing lapses, mistakes and violations, rather than just concentrating on slips. Looking at becoming lost as a form of error provides a unique perspective on day-to-day errors in the wild. Finally, a potentially much more fruitful approach to the problems of human error in socio-technical environments is Dekker's [8] suggestion that errors are most malleable when approached not as human errors but as systemic events. This perspective supports a functional active approach to errors rather than just encouraging end-users to "follow standards". The traditional approach typically relies on training and worker compliance rather than a principled approach which is more likely to make a robustly safe system.

Technique for Human Error Rate Prediction (THERP) is a frequently applied method for predicting human error rates and for evaluating the degradation of a human-machine system likely to be caused by human errors in association with factors such as equipment reliability, procedures, and other factors [12]. A user of THERP has to estimate the probability that an error will occur and the probability that the error will lead to system failure in order to calculate the probability that a class of errors will lead to system failure and the total system or subsystem failure rate.

Becoming lost and getting found literature provides a wealth of best practices [11], albeit typically being focused on being lost in wilderness. However the collected experience in how-to-handbooks about urban and special populations can provide insights into construction of mitigation plans. An interesting example of this is that Alzheimer's sufferers that wander away almost always go in a straight line [11], so wandering is not really wandering as we might imagine.

Contributing to understanding the process of both creating a route and mitigation efforts are the various studies on elders and special populations doing urban navigation [2-4, 13]. These have provided insight into how people get lost on a route and what helps them when lost.

## Distributed cognition

In most traditional approaches, *human cognition* has been seen as existing solely "inside" a person's head, and studies on cognition have often disregarded the physical and social surroundings in which cognition takes place. *Distributed Cognition* [9, 10, 17] provides an effective theoretical framework for understanding what humans can achieve and how artefacts, tools, and socio-technical environments can be designed and evaluated to empower human beings and to change tasks. Applying this framework to guided route traversal and errors presents new and unique challenges, and in return it will create a deeper understanding of distributed intelligence.

Any discussion of error in a modern context must also involve distributed cognition (which is also discussed as distributed intelligence) [16]. Tasks that use computa-

tional assistance in planning and/or executing a task depend on the distribution of the cognitive act across an individual and the system. In the case of ASSISTANT the navigational acts are spread across the senior user and her Personal Navigation Device (PND) [10]. So when an end-user comes to the end of a trip segment, she is alerted to the need for exiting from the bus (or metro) by multimodal alerts (vibration, sound and/or flashing screen on the PND); similarly the series of waypoints to navigate from the last public transportation to the final goal of the route can be read by the end-user off of the PND screen.

So, then, the process of mitigation of the lost state (or error condition as we shall refer to in the rest of the paper) becomes a question of how to most effectively provide guidance via PND and user. Why is this important? By understanding how the user builds her understanding of the process with the output of the PND, the system can reliably bring the traveller back on track, or appropriately summon help.

## **ASSISTANT, the platform and the scope of error problem**

ASSISTANT is an AAL project that provides accessible support for the use of public transportation, seamlessly spanning trip planning via an online application, scheduling and providing guides for multi-step journeys, including intermodal and trans-national transfers with provision for supplying information to help the end-user from the last transit stop to the final destination. The target group is older people, particularly those travelling to novel places or beginning to use public transportation after losing ability to drive themselves. The design of the system will be accessible and may be of use to travellers new to an area, young people, and those with accessibility problems. ASSISTANT builds its technology on familiar interfaces, e.g. the PC, tablets, and smart-phone, providing only relevant information at the right time and in the right fashion via audio, visual and haptic cues. ASSISTANT provides safety by error detection and remediation. The system identifies the vehicle to board and signals to the user its arrival, and reminds the user when to exit. ASSISTANT predicts or anticipates the users and the public transportation vehicle position when location data is missing; the positioning of both the user and public transport vehicles will be estimated using a prediction function.

Being lost is an experience common to anyone who has travelled. However, for the intended users of the application becoming lost is more daunting. Working with error is in two parts: trapping an error, that is to say identifying that the system is in an error state and then identifying the characteristics of the error. Having 1) flagged an error state and done 2) analysis of the attributes of the error, then the second part of the error processing is generating a mediation of the error state. Using the mediation plan the system will attempt to either 1) get the user back on track to get to the routes goal or 2) summon appropriate help to resolve the lost state.

To begin designing an error trapping and mitigation sub-system, first the problem space needs to be estimated. In this case the space is every possible error that could be trapped. Estimating the number of potential errors in any human endeavour could be a daunting task. One way to look at it is that the number of errors is the factorial of the

count of each individual possible error, a very large number. This is the enumeration of the things that *could* go wrong. But is it really that bad? There is a difference between potential and probability. In taking the bus it is *possible* that the bus is driven into the sea, but the probability of it doing so is very low. Our initial studies with seniors using public transport reported that the list of errors in a supported route is short. As in so many other human activities the sorts of things that typically go wrong is fairly small [15] but covers a majority of instances (a case of something like the classic *Pareto principle* [1]). Assuming that the error distribution in seniors' use of public transportation is similar to the examples in the literature (an assertion that we will be studying as the project continues), we can then derive two broad categories of trapped errors. This can be done by dividing the number of errors into two sets: ones that are known **and** which something can be done about and ones not known (or can not be identified beyond that it *is* an error) and/or the system can not do anything about.

## Error capturing

ASSISTANT has a finite number of sources of information about the route that the user is taking, and thus determining if there is an error state. Table 1 enumerates the sources of information and what reasonable conclusions could be drawn. The paucity of types of data and informational states they could provide can nevertheless provide reliable information about the user on the route.

Table 1 – Sensors and Information

Sensor /information	Information collected	Deduction about data
GPS	Location	Off of route, not making progress
Route path state	History of nodes traversed and timing	'Rocking' over two states $\Rightarrow$ not sure of context
PND compass	Orientation of PND	Relation to POV <sup>1</sup> maps
PND alive on network (ping)	PND on network and active	Route guidance not blocked by: 1) dead PND and 2) off of network
PND alive in cell system	PND has phone capability	Panic button works, also system can call end-user with verbal instructions
PND accelerometer	Velocity and acceleration (typically in 3 axes)	Speed of user movement
User interface controls	End user's choices from PND menus & buttons	What the end user desires

ASSISTANT's error model is a first approximation of a rule-based error capturing and mitigation system. In the case of this project we circumvent the more complete approach of extracting the existence and the type of error based on user and system

<sup>1</sup> POV maps – Point Of View maps, the kind where you have the typical 'You are Here' dot and label *and* the orientation of the map assumes that you are oriented in the same direction as the map.

behaviour. Partially this is because the data is so sparse, and partially because the error model has to be in effect before the initial use of the system. Use data and system state will provide, over the long run, data enough to calibrate the error model. We expect the types of errors to be quite constrained and the amount of data to be sparse, a difficult combination. Beyond that, the capturing of error has a high enough risk (which can be thought of as risk as the product of probability and cost, where cost is of course to be understood in a broad sense), in terms of both type 1 and type 2 confusion matrix types, that we have chosen to make it rule based and thus more reliable. From the perspective of user adoption it is preferable to err in the direction of false negative (type 2) than type 1 because a series of false positives (flagged error states) is highly likely to result in system abandonment. The design of the error trapping and mitigation parts of ASSISTANT will be modular enough so that other error engines which are data driven could be inserted in to augment the rule based one.

Any route being traversed in the ASSISTANT system will be constantly monitored by the error module. As information arrives (typically it will be in the form of updates of user position and some system information, like battery state) a state machine will be built up and the error analysis module will be matching the current state and the set of states and their information with an error signature. When there is a match, the module interacts with the system and typically with the user's PND, although in the case of PND being out of power or not responding for a selected time, the system will take action using the network and phone system as part of the mitigation strategy.



Figure 1 - Concept map of ASSISTANT's error model

Initially error trapping will be rule based and extended by user modeling data as the system is used enough to gather data sufficient to classify error states. These initial error rules (see Figure 1) are based on internal PND states, network status, PND location, whole system analysis, and user interaction. As user data is accumulated

over many trips, successful and with errors, the error-trapping engine builds up probable error signatures for this user and uses these to identify and classify errors.

## Error classification

As we have seen in the section on the scope of the problem, the types of errors is typically small and we are basing our classification on the assumption that this holds true also for the types of errors that ASSISTANT users will encounter. Our initial interviews with seniors in San Sebastian, Paris and Vienna supported this assumption. When we start field trials later we will be spending particular attention to the types and number of instances of the navigation errors we will discover. A part of the early trials of the prototype system will be capturing and analysing user errors to validate this claim, that typical travel errors follow Pareto's division (80% of *instances* of errors are caused by 20% of *types* of errors). If we can provide simple workarounds for a large number of those 20% common errors we are well onto the way of dealing with 80% of the cases of error. An example of this is the common occurrence of not getting off the bus at the correct stop, for this there may be any number of reasons why this happened (falling asleep, becoming distracted), but the point here is that the location information about the user is not corresponding with where she should be but rather with the bus as it continued on its route.

Table 2 Error types, sources and tests

Type	Source	Test and what you know
Internal PND state	Battery	Battery very low, short time available to mitigate
Network status	No signal (turned off, broken, battery dead, stolen, lost)	You know: a. Time lost signal b. Location lost signal
PND off route	Between where smartphone is and where it should be	You know: 1) Time 2) Stage of trip 3) In bad part of town?
	In path but no movement for more than X time	
	GPS failure (phone connected no location information)	
Panic button pushed	User	You know: a) Where user is and b) What route they were on
Whole system analysis		Cannot get to first bus/mobile/goal on time
		Anticipated arrival to bus/metro/transfer too late (after bus/tram left)

The ASSISTANT user model will contain settings for highly tailorable user interfaces and system interaction styles preferred by each user. Depending on the user model, the PND may recommend exiting and crossing the street to take the bus one stop back *or* it may summon help if the person is particularly fragile. For the other 80% of possible errors the system brings in, using the PND's phone functionality, a person in the end-users social network.

Trapping an error as we have discussed above starts the process of error mitigation: once the error has been flagged then the system classifies the type of error. The type of error explicitly describes the error and makes no attempt to deduct the cause of error. Typical error classification types might be low battery, no response from phone ping, PND not on route, panic button pushed, or too late for next transfer. Along with the classification the system sends contextual information to the error mitigation module.

There are some other user modelling parameters that apply to both error capturing and classification. The granularity of the check for error state depends on the granularity of the local transport data update. The compensation strategy for lost GPS signal depends on context and locale. An example of this is when the metro station is underground and the PND is shielded from recovering GPS signals. Another example of error parameters is the chain of persons / phone numbers that each traveller has: they may start at their caregiver, but in the case that the caregiver cannot indicate that she will take responsibility for the situation, by entering a code in the phone, the next step may be other caregivers incrementally leading up to an emergency number (i.e. 119) or may jump directly to the emergency number.

## Error mitigation

The error mitigation class will be built up as ASSISTANT user trials collect data on mitigation. There is decision tree support in mitigation class, not unlike diagnosis decision trees in classical automatic medical diagnosis engines. The parameter class has a subclass named mitigation theme that supports generic types of mitigation strategies [18] allowing easier configuration to the individual user, and also allowing very tailored mitigation schemes.

Table 3 Error context and classifications

Type	Context	Strategy
Panic button pressed	Machine state / Route state PND state / Mitigation theme	Cascading alarm Mitigation capacity
Got off at bus stop after goal	Bus / metro route(s) /System type <sup>2</sup> / PND state/Mitigation theme	Cascading alarm Mitigation capacity
Transfer error	Route / PND state Mitigation theme	Mitigation capacity // Decision tree: 1. Forward 2. Is next one to go back a good choice 3. If not is there an alternate route 4. If not go home (and cascading alarm) 5. Back 6. Go home

Specific error mitigation strategies are tailored to each user. They may a) push mitigation actions to user, b) connect the user to a helpful person, or finally c) summon human emergency aid.

<sup>2</sup> In D-bus (San Sebastian, Spain) average stop distance from previous is quite short and several interviewed seniors said that they 'would just walk back to the stop.'

Table 4 - Error Mitigation Themes

Error classification	Battery very low	No PND response from	Missed stop (one stop too far)	Missed stop (2+ stops too far)	Panic button pressed
Theme					
Old	Vibrate and warning to contact driver	Alert Caregiver	San Sebastian: walk back / Paris: take bus/metro on stop back	Take bus/metro <many> stops back	Escalating Summon caregiver
Fragile old	Summon caregiver	Escalating Summon caregiver	Summon caregiver	Summon caregiver	Escalating Summon caregiver
Child	Summon caregiver	Escalating Summon caregiver	Summon caregiver or Alert caregiver and do typical mitigation		Escalating Summon caregiver
Typical	Vibrate and warning	Nothing	San Sebastian: walk back /Paris: take bus/metro one stop back	Take bus/metro <many> stops back	Connect with transportation help line
Mobility impaired	Vibrate and warning	Warn caregiver	Depends on stop accessibility annotation	Depends on stop accessibility	Escalating Summon caregiver
Visually impaired	Vibrate and auditory warning	Warn Caregiver	San Sebastian: walk back /Paris: take bus/metro one stop back	Take bus/metro <many> stops back	Escalating Summon caregiver
Hearing impaired	Vibrate and visual warning	Warn Caregiver	San Sebastian: walk back/Paris: take bus/metro one stop back	Take bus/metro <many> stops back	Escalating Summon caregiver
Custom mitigation	Tailored response depends on needs and abilities				

The error trapping and mitigation class tree will be instantiated as experience in what works and what is important is gathered in user trials. Mitigation strategies will also take into consideration the dynamic context that the error arises in. Table 5 shows the context and strategy that need to be taken. Table 4 shows some error mitigation themes and details.

## Implementation

Parts of the proposed system have been implemented in another system, MAPS [5], which one of the authors has worked on. The interface to generate an error test and tie to monitored events was part of the MAPS system. The idea of collecting user accessibility needs and preferences was explored in another of the author's projects, the EU4All project [6].

Figure 2 shows a design that supports attaching an error test to a step in a task support script provided to young adults with cognitive disabilities [7]. These preliminary prototypes support for error trapping and mitigation required explicitly declared tests; in the current paper we advocate a more general and universally applicable error trapping engine approach.

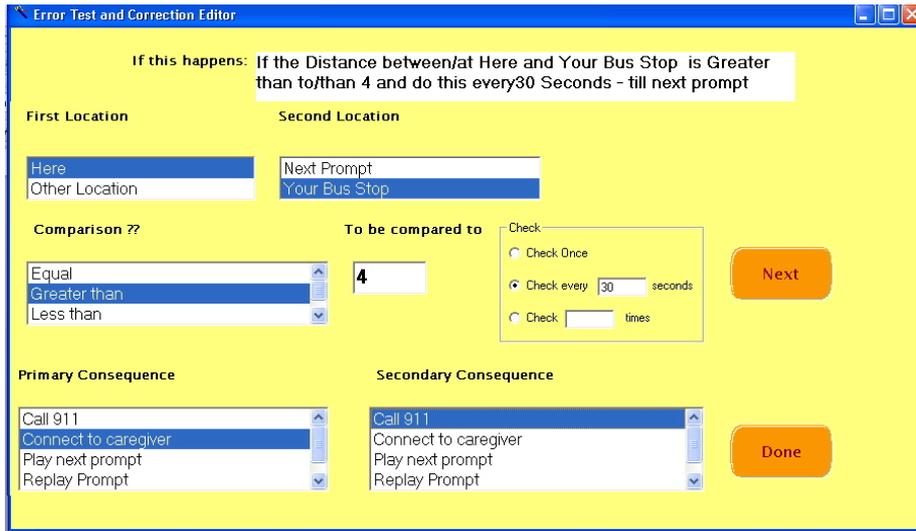


Figure 2 Error capture conditions

## Further work

The ASSISTANT project, structured in a triple tiered spiral development approach, will provide a container for proposing and refining this approach to error trapping and mitigation. Most interesting will be the idea of mitigation types (which make error systems free from individual configuration, at least from the bottom up) and error type classification which can lead to easier implementation and broader acceptance. These will be tested in the user trials in San Sebastian, Paris and Vienna.

## Acknowledgments

The authors wish to thank the AMBIENT ASSISTED LIVING JOINT PROGRAMME, which provided the support for the ASSISTANT project from which much of this work emerged. The Centre for Lifelong Learning and Design at the University of Colorado provided initial studies for designing systems for task support. The EU4ALL project, funded by the European Commission, gave inspiration through distance learning personalization; and most notably in that project Carlos Velasco of the Fraunhofer Institute for Applied Information Technology (FIT). We would also like to thank our fellow researchers in the Health Division of TECNALIA.

## References

1. Anderson, C., The Long Tail. *Wired*, 2004. 12(10).
2. Barbeau, S., Georggi, Nevine Labib, Winters, Philip Travel Assistance Device Deployment to Transit Agencies, 2010, State of Florida Department of Transportation: Tallahassee, Florida, USA.
3. Bradley, N.A., Dunlop, Mark D. , An Experimental Investigation into Wayfinding Directions for Visually Impaired People. *Personal and Ubiquitous Computing*, 2005. 9: p. 395-403.
4. Bryden, K., Charlton, J., Oxley, J., Lowndes, G., Changes in wayfinding ability with age: Self-restrictions, co-piloting and the use of navigation systems, in *The 12th International Conference on Mobility and Transport for Elderly and Disabled Persons (TRANSED)*, 2010: Hong Kong.
5. Carmien, S., MAPS: creating socio-technical environments in support of distributed cognition for people with cognitive impairments and their caregivers, in *Extended abstracts of the 2004 conference on Human factors and computing systems*. 2004, ACM Press: Vienna, Austria. p. 1051-1052.
6. Carmien, S. and A.M. Cantara, Diagnostic and Accessibility Based User Modelling, in *User Modeling and Adaptation for Daily Routines Providing Assistance to People with Special Needs*, E. Martln, Haya, P.A., Carro, R.M., Editor. 2012, Springer. p. 232.
7. Carmien, S., Fischer, G. Design, Adoption, and Assessment of a Socio-Technical Environment Supporting Independence for Persons with Cognitive Disabilities. in *ACM Conference on Computer-Human Interaction CHI08*. 2008. Florence, Italy.
8. Dekker, S., *The field Guide to Understanding Human Error*. 2006, Burlington, VT: Ashgate. 236.
9. Fischer, G., Distributed Cognition: A Conceptual Framework for Design-for-All, in *Proceedings of HCI International 2003*, C. Stephanidis, Editor. 2003, Lawrence Erlbaum Associates, Mahwah, NJ: Crete, Greece, June 2003. p. 78-82.
10. Hollan, J., E. Hutchins, and D. Kirsh, Distributed Cognition: Toward a New Foundation for Human-Computer Interaction Research, in *Human-Computer Interaction in the New Millennium*, J.M. Carroll, Editor. 2001, ACM Press: New York. p. 75-94.
11. Koester, R.J., *Lost Person Behavior*. 2008, Charlottesville, VA: dbS Productions.
12. LaSala, K.P., *A Practical Guide to Developing Reliable Human-Machine Systems and Processes*, in RAC publication, 2002.
13. Müller, C., Wan, Lin, Hrg, Dalibor, Dealing with Wandering: A Case Study on Caregivers' Attitudes towards Privacy and Autonomy when Reflecting the Use of LBS, in *Group 102010*, ACM: Sanibal Island, Florida, USA. p. 75-84.
14. Norman, D.A., Design Rules Based on Analyses of Human Error. *Communications of the ACM*, 1983. 26(4): p. 254-258.
15. Reason, J., *Human Error*. 1990, Cambridge, UK: Cambridge University Press.
16. Rogers, Y. *An Introduction to Distributed Cognition*. 1997.
17. Salomon, G., ed. *Distributed Cognitions: Psychological and Educational Considerations*. 1993, Cambridge University Press: Cambridge, United Kingdom.
18. TAP, Synthesis of Real Traveller Information Needs within Intermodal Journeys, in *CODE TR1103 - Accompanying Measure of the Transport Sector of the Telematics Applications Programme (TAP, 4th Framework Programme for RDT&D, 1994-1998)*, 2000.