

Graphical interface for eCall incidents

Claudiu Dan BARCA

The Romanian-American University, Bucharest

Using data on incidents eCall in traffic control center I have designed a graphical interface between PSAP 112 and national traffic control center. The interface is modular, each component is implemented as a separate module, of which architecture respects the design pattern MVVM (Model View View-Model). This will get a timely response for traffic management in the area and obtain traffic information in the area.

Keywords: *eCall, graphical interface, traffic management, traveler information services, Model View View-Model*

1 Introduction

The single emergency service, call 112, was conducted in Europe. This is a practical way to make the free movement of European citizens to safer. Important efforts have been made in the European Union that, in case of crash, 112 can be automatically dialed from inside the car, In this context all new cars should be equipped with the automatic emergency call system eCall - Emergency -eCall [1,2,3]

The instalation of the ECall based on the service 112 in types of vehicles which are to be manufactured after March 31, 2018 should be promoted to increase the degree of system implementation. Regarding the types of vehicles type-approved before 31 march 2018, the subsequent installation of an Ecall system is possible [4]

There are serious and very serious accidents when people's lives depend on alarm systems such as eCall [5,6,7]. The service is automatically activated and calls the European emergency number 112 by transmitting information about the accident to dispatch 112 [8]. The eCall system allows the generation of a 112 emergency call and sends a set of data on the vehicle, where the accident occurred, etc. [9]. The request can be performed manually or automatically [10,11]. When there is no data necessary for solving the case, it can appeal to a European database.

eCall is a European Commission project that will carry out road safety through rapid intervention specialized services in case of an accident anywhere in the European Union.

ECall is an open system platform, will send other useful information to beneficiaries such as traffic management centers. These are based on logging the incidents submitted by eCall platform to inform road users and other media communications on the incident location to avoid congestion and affected accidents routes. In this context we designed a graphical interface that allows data exchange between PSAP 112 and the national traffic management center.

2. Design interface

I used Composite Application Library [12] to design the interface, which helps architects and developers to develop applications with Windows Presentation Foundation (WPF). Composite WPF applications consist of separate "tracks", fully functional, which creates a single unified user interface. Composite Application Library accelerates composite application development using design patterns to help them.

Composite Application Library is designed to address the demands from developers and architects that create WPF client applications and need to solve the following:

- Developing clients composed of independent but cooperative modules.
- Separating development modules from a

framework for integration (shell framework); In this way teams can focus on developing specific modules but not the entire architecture WPF

- Using an architectural framework for embedded applications to produce consistent and high quality results.

3. Interface description

The interface messaging application eCall (Figure 1) is divided into three distinct areas (called regions): the upper - the toolbar (Type Toolbar / commandbar) area, the center - contains components the main interface (Overview, message Board, Search) you can select independent by changing tabs

(type Content Tab) and the right - the quick access to important messages / urgent (watchlist) and statistics (statistics), each of these two are separately encapsulated in an expandable list (Expander ViewList).

Interface Overview. The application starts with reports on the status of all messages (Message Status Report Chart) and the intervals of the faults (Chart incidents Report). The first graph shows the number of messages received (incoming), completed (finalized) or ongoing / settlement (pending). The second graph shows the number of incidents per time of day, to detect the most dangerous period (with the highest risk level).

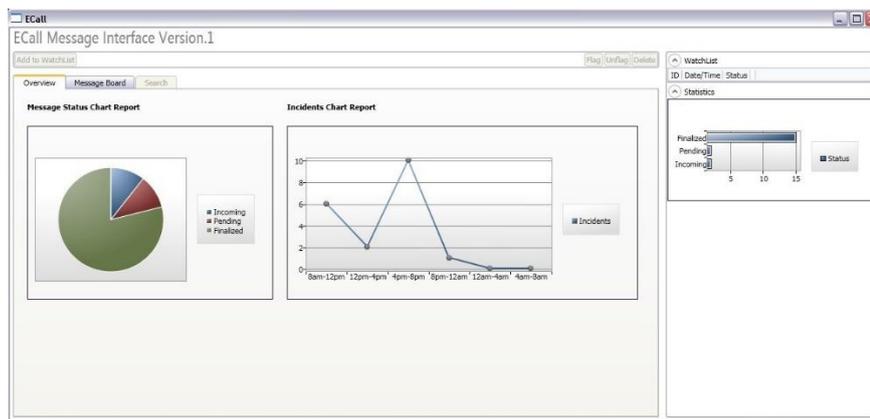


Fig. 1 – ECall interface messaging application (overview)

Component selection can be changed by another component, such as Table messages (Message Board) shown in (Figure 2). In the right will always appear the chart with statistics on the

current number of active incidents (pending receipt or pending scroll / settlement) and the list of important messages that require supervision (Watch list).

| ID | Date/Time | Index | Incident address | Description | Status | MSD | Flag |
|----|-----------------------|-------|---|---|-----------|-----|------|
| 1 | 6/15/2012 12:37:31 PM | S345 | Bldv Gloriei, Nr. 101-103 | Accident 2 autoturisme, nu sunt raniti | Incoming | | |
| 2 | 6/15/2012 12:35:31 PM | S344 | A1, Bucuresti-Pitești, km 20 | Accident autocar, blocul ruber | Incoming | | |
| 3 | 6/15/2012 10:57:31 AM | S343 | A2, Bucuresti-Constanta, km 110 | Accident usor, autoturism sedan categoria B, nu sunt raniti | Pending | | |
| 4 | 6/15/2012 10:32:31 AM | S342 | Calea Serban Voia, Nr.132-134 | Accident de motocicletă cu autoturism categoria B, 1 ranit | Pending | | |
| 5 | 6/14/2012 10:32:31 PM | S341 | A3, Brasov-Onadea, km 156 | Accident 5 autoturisme categoria B, 1 decedat, 4 raniti | Finalized | | |
| 6 | 6/14/2012 8:12:41 PM | S340 | A2, Bucuresti-Constanta, km 33 | Accident 1 autoturism categoria B, intrat in parapet, 2 raniti | Finalized | | |
| 7 | 6/14/2012 8:09:03 PM | S339 | DN1, Localitatea Stajera (Alba Iulia - Turda) | Accident 2 autoturisme categoria B, 5 decedati | Finalized | | |
| 8 | 6/14/2012 7:42:14 PM | S338 | DN1, E60, Negreni | Accident autocarion - autoturism categoria B, 2 decedati | Finalized | | |
| 9 | 6/14/2012 7:17:31 PM | S337 | DN1, 450+075 - 451+075, Copaceni | Accident motocicletă, 1 decedat, 1 ranit (copasager) | Finalized | | |
| 10 | 6/14/2012 6:57:01 PM | S336 | DN1, 460+466 - 461+465, Intre Martinești si Valcele | Accident autocarion, derapat de pe caroseabii, niciun ranit | Finalized | | |
| 11 | 6/14/2012 6:21:11 PM | S335 | DN1, 462+118 - 463+117, intrarea in Valcele | Accident autoturism sedan categoria B, pierdere control volan, 1 ranit | Finalized | | |
| 15 | 6/14/2012 6:11:19 PM | S334 | DN1, 471+000 - 472+000, Intre Feteac si Cluj-Napoca | Accident 2 autoturisme categoria B, 1 ranit usor | Finalized | | |
| 25 | 6/14/2012 5:51:29 PM | S333 | DN1, 482+800 - 482+800, Intre Cluj-Napoca si Floresti | Accident autocarion, pierdere control, iestire de pe caroseabii pe contrasens, 2 raniti | Finalized | | |
| 35 | 6/14/2012 5:43:55 PM | S332 | DN1 491+481 - 492+480, intrarea in Gilau | Accident usor autoturism categoria B, defectiune volan, intrare parapet, niciun ranit | Finalized | | |
| 45 | 6/14/2012 5:13:16 PM | S331 | A1, Arad-Timisoara, km 12 | Accident motocicletă, depasire de viteza, pierdere control, 1 decedat | Finalized | | |
| 55 | 6/14/2012 3:12:09 PM | S330 | A1, Bucuresti-Pitești, km 41 | Accident autoturism categoria B, viteza + derapare pe caroseabii unice, 1 decedat | Finalized | | |
| 65 | 6/14/2012 1:50:09 PM | S329 | A1, Bucuresti-Pitești, km 66 | Accident 2 autoturisme, intrare pe contrasens din cauza pierderii controlului, 4 raniti | Finalized | | |
| 75 | 6/14/2012 11:15:00 AM | S328 | A2, Bucuresti-Constanta, km 116 (Carnavada) | Accident 3 autoturisme, depasire viteza, 3 raniti, 1 decedat | Finalized | | |

Fig. 2. Messages table

Within the Message Board component, we find all messages sorted by lowest ID. We can mark / unmark / delete / add a message to our watch list (watch list) or more, through their normal selection and proper actuation. For example, if a message has been selected and sent to the watch list, will appear in the appropriate box. Subsequently, if the message no longer needs necessary supervision, it can be deleted (even the table posts, but obviously this is just an interface, there is a deletion in the database).

Each message in the Messages table is accompanied by a package MSD (Minimum Set of Data) which can be viewed by the action command View from desired message. A window will open (Figure 3) where we have information on date / time, location, direction (GPS coordinates) and last known locations. Also, within this window we can locate the incident location using an interactive map to locate the action (Locate incident).

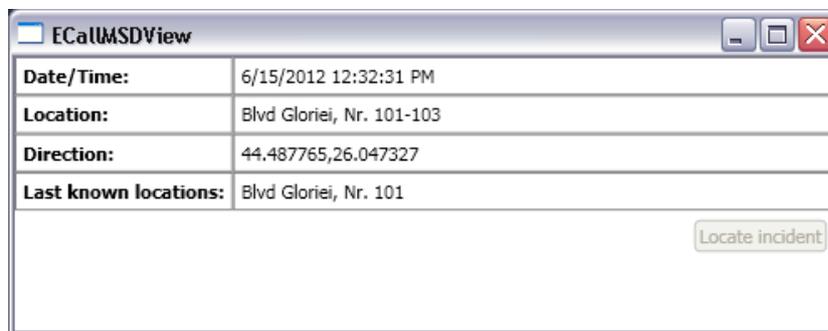


Fig. 3. View information MSD

Possible actions:

1. Change component
2. Cloning MessageBoard message to the watchlist (Add To Watchlist)
3. Delete the message from the message board or watchlist
4. Filter according to selectable options
5. Select / deselect messages
6. Select / deselect all messages
7. Mark / unmark (Flag / Unflag) posts
8. View MSD (per message)
9. Expand list in the right area

4. Interface components

The interface has a modular design, where each component is implemented as a separate module. These modules are loaded at runtime (using the advantages of .NET Framework) along with the data model used to describe the messages. The communication between modules is achieved through a range of services available to all modules [12,13].

The application's architecture respects the design pattern MVVM (Model View View-Model) which means that each module has three separate levels: model-data (Data Model - common to each mode), view-model (View-Model - way a parse and identify how data model specific to each module), graphical view (View - way data is represented graphically model for user interaction).

The interface can identify six components / modules: General Overview, Message Board, Search, Watch List, Statistics and the Toolbar. The data model behind the visual interface describes all the content of a message eCall transmitted to the central server (abstracts a message eCall) and notifies any changes made to it (is an observable data model - any change in the model will synchronize and will reflect the changes). Each module interprets this data model differently, depending on its purpose and function performed by the View model (ex. the Statistics module uses only the

number of posts under three conditions mentioned above).

Model data

The database is accessible through service modules. These services inspect public properties exposed by the model, thus filling in the description field values. The data, as I mentioned, is observable and the interface notifies when changes occur (data model implements INotifyPropertyChanged interface exposed in the .Net Framework). When you start the application, the data model is initiated with content. So we will have a list of messages where each message is defined by the data fields present in the model. When we interact with the data model we actually interact directly with the content of the messages. To change the content of a message, we call all the properties exposed by model. Once we have achieved the desired change, the property will notify this to the interface, at which time it will reset the level of View.

The data model has the following fields (exposed via public property):

id: ID of the message is the registration number in the database

- associated property: ID
 - Time: the time and date the message was recorded
- associated property: Time

- Index: Index message Event index (derived from the database of incidents, selected by the operator)

- associated property: Index
 - incidentAddress: address on the scene
- associated Property: IncidentAddress
 - freeText: text space for notes, incident description
- associated property: FreeTex
 - status: status message - resolved, pending, ongoing allocation
- associated property: Status
 - flag: marking message
- associated property: Flag
 - MSDtime: time and date on which the incident occurred
- associated property: MSDTime
 - MSDlocation: address details about the incident, GPS positions
- associated property: MSDLocation
 - MSDdirection: directions to the place of incident (position before the crash- GPS coordinates)
- associated property: MSDDirection
 - MSDlastLocation: new locations (the last two, three GPS position) - associated property: MSDLastLocation

Components / Modules

- General Overview (Figure 4)

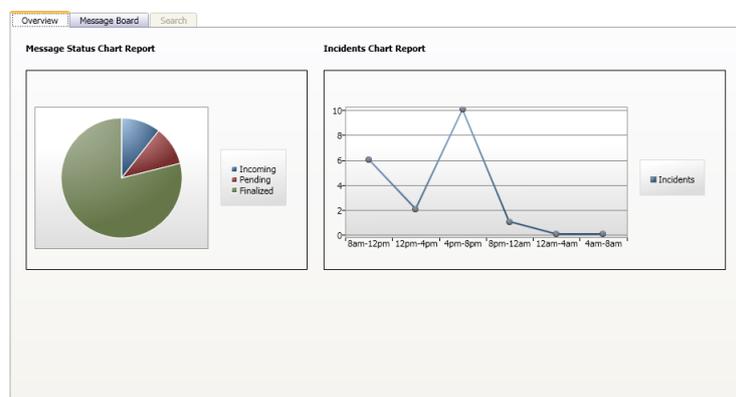


Fig. 4. Components / modules (overview)

This module exposes two graphs / reports on the number of posts, under three states (pending allocation under finalization / settlement and completed) and the number of incidents in a 11 intervals of a day (24h). These calculations are performed only on messages found in the table, so archived messages will not be considered (only local posts).

- Message Board

The message board (Figure 5) lists all the messages that the user wants to view. Each message is defined in the data model, the fields are visible in the form of table heads. Only the MSD (minimum set of data) is visible separately when choosing to View the selected message. At the bottom we have two buttons - selection of all messages or reset selection, and two filtering options - after status (Incoming, Pending, Finalized) or only marked messages (Flagged).

| ID | Date/Time | Index | Incident address | Description | Status | MSD | Flag |
|----|-----------------------|-------|---|---|-----------|-----|------|
| 1 | 6/15/2012 12:37:31 PM | 5345 | Blvd Gloriei, Nr. 101-103 | Accident 2 autoturisme, nu sunt raniti | Incoming | | |
| 2 | 6/15/2012 12:35:31 PM | 5344 | A1, Bucuresti-Pitesti, km 20 | Accident autocar, blocaj rutier | Incoming | | |
| 3 | 6/15/2012 10:57:31 AM | 5343 | A2, Bucuresti-Constanta, km 110 | Accident usor, autoturism sedan categoria B, nu sunt raniti | Pending | | |
| 4 | 6/15/2012 10:32:31 AM | 5342 | Calea Serban Voda, Nr.132-134 | Accident de motocicletă cu autoturism categoria B, 1 ranit | Pending | | |
| 5 | 6/14/2012 10:32:31 PM | 5341 | A3, Brasov-Oradea, km 156 | Accident 3 autoturisme categoria B, 1 decedat, 4 raniti | Finalized | | |
| 6 | 6/14/2012 8:12:41 PM | 5340 | A2, Bucuresti-Constanta, km 33 | Accident 1 autoturism categoria B, intrat in parapet, 2 raniti | Finalized | | |
| 7 | 6/14/2012 8:09:03 PM | 5339 | DN1, Localitatea Stajenis (Alba Iulia - Turda) | Accident 2 autoturisme categoria B, 5 decedati | Finalized | | |
| 8 | 6/14/2012 7:42:14 PM | 5338 | DN1, E60, Negreni | Accident autocamion - autoturism categoria B, 2 decedati | Finalized | | |
| 9 | 6/14/2012 7:17:31 PM | 5337 | DN1, 450+075 - 451+075, Copaceni | Accident motocicletă, 1 decedat, 1 ranit (copasager) | Finalized | | |
| 10 | 6/14/2012 6:57:01 PM | 5336 | DN1, 460+466 - 461+465, intre Martinesti si Valcele | Accident autocamion, derapat de pe carosabil, niciun ranit | Finalized | | |
| 11 | 6/14/2012 6:21:11 PM | 5335 | DN1, 462+118 - 463+117, intrarea in Valcele | Accident autoturism sedan categoria B, pierdere control volan, 1 ranit | Finalized | | |
| 15 | 6/14/2012 6:11:19 PM | 5334 | DN1, 471+000 - 472+000, intre Feleac si Cluj-Napoca | Accident 2 autoturisme categoria B, 1 ranit usor | Finalized | | |
| 25 | 6/14/2012 5:51:29 PM | 5333 | DN1, 481+800 - 482+800, intre Cluj-Napoca si Floresti | Accident autocamion, pierdere control, iesire de pe carosabil pe contrasens, 2 raniti | Finalized | | |
| 35 | 6/14/2012 5:43:55 PM | 5332 | DN1 491+481 - 492+480, intrare in Gilau | Accident usor autoturism categoria B, defectiune volan, intrare parapet, niciun ranit | Finalized | | |
| 45 | 6/14/2012 5:13:16 PM | 5331 | A1, Arad-Timisoara, km 12 | Accident motocicletă, depasire viteza, pierdere control, 1 decedat | Finalized | | |
| 55 | 6/14/2012 3:12:06 PM | 5330 | A1, Bucuresti-Pitesti, km 41 | Accident autoturism categoria B, viteza + derapare pe carosabilul umed, 1 decedat | Finalized | | |
| 65 | 6/14/2012 1:50:09 PM | 5329 | A1, Bucuresti-Pitesti, km 66 | Accident 2 autoturisme, intrare pe contrasens din cauza pierderii controlului, 4 raniti | Finalized | | |
| 75 | 6/14/2012 11:15:00 AM | 5328 | A2, Bucuresti-Constanta, km 116 (Cernavoda) | Accident 3 autoturisme, depasire viteza, 3 raniti, 1 decedat | Finalized | | |

Fig. 5. Table messaging

When we select a message, or more, and choose to add them to the watch list (Add To Watch List), these will be added (as cloned messages) in the appropriate box (Watch List) located in the right region. The message board can be monitored throughout the entire activity as they are notified in case of data content changes

(ex: change the message with ID 2's status from Finalized to Pending) and therefore the behavior is observable in real time. Thus, through this list, we can isolate high priority messages and monitor them separately. If we want to delete messages from the list we just do it by pressing X, colored in red (Figure 6)

| ID | Date/Time | Status | |
|----|-----------------------|-----------|---|
| 5 | 6/14/2012 10:32:31 PM | Finalized | X |
| 4 | 6/15/2012 10:32:31 AM | Pending | X |
| 3 | 6/15/2012 10:57:31 AM | Pending | X |
| 1 | 6/15/2012 12:37:31 PM | Incoming | X |

Fig. 6. Watch List

- Statistics

Statistics module (Figure 7) shows the number of messages that are in the three states (completed, pending,

pending allocation) as a bar graph. This graph is always visible in the right region and is also notified when changing data content (ex.: longer resolve an incident).

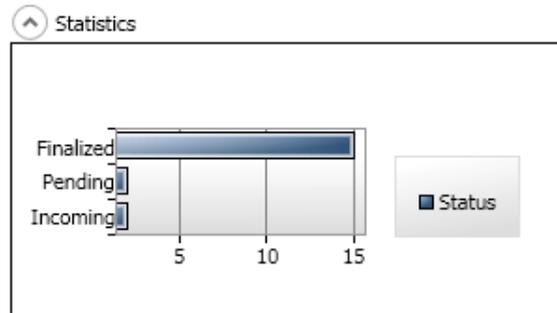


Fig. 7. Statistics module

- Toolbar can act on a message in Message Board (Figure 8)

ECall Message Interface Version.1



Fig. 8. Toolbar

- Add to WatchList
By selecting a message or several messages and using this option, we are copying those messages in the watch list (right list, the shortcut).
 - Mark (Flag)
By selecting a message or several messages from the board and using this command, we will be able to mark those messages through a red warning sign.
 - Unmark (Unflag)
Mark is the reverse order, removes the selected messages' warning sign.
 - Delete
By selecting a message or several messages making and using this command, we will be deleting messages. Erasure occurs only at the interface level; messages remain in the database and can be part of the archive. Every command is implemented independently, in this way using a design-pattern type Command, and command action is a method call made by the mediator - contains services implemented at the application level that can manipulate data content.
- The interaction between modules is achieved through a mediator containing a list of services implemented at the application level that can manipulate data content:
- *Message XML Reader*- Reads and parses the data stream coming from the database; stores into a collection data.
 - *Message Control Service*, implements the following functionality on data collection:
 - *Set Message Board Collection*: this functionality is achieved through a connection (data binding) with the data collection that contains instances of the data model complemented with information obtained from flow database.
 - *Add To WatchList* : this functionality assigns a message to the watchlist,
 - *Flag/Unflag Message*: This functionality is assigned to the command Flag / Unfla - in toolbar
 - *Delete From Message Board*: this functionality is assigned to the Delete command - the command bar
 - *Delete From Watch List*: this functionality is assigned to the command "X" associated with an element of watch list
 - *StatisticsService*: Based on the collected

5. Interaction between modules

data, the statistical/analytical computation is needed to build the graphs of the two components, Overview and Statistics. This service has methods (visible from outside mediator) to call the calculation of statistics on new data and get those calculations to build graphs.

6. Conclusions

ECall application messaging interface allows quick access to important messages / emergency type eCall and statistics.

The data model used in the design of the interface shows all the content of a message sent by the eCall PSAP 112 and notify any change is made on sa. In conclusion, we get useful data in real time on the incident eCall

References

- [1] Safety Forum- The eCall Project- eCall Expert Meeting Helsinki April 2005
- [2] MEMO/13/547- European Commission- Brussels, 13 June 2013
- [3] Dietmar Zlabinger- Questionnaire eCall/eCall-Flag-. Mobilregulierungsdialog 2012-03-23
- [4] George Căruțașu, Cezar Botezatu, Mihai Botezatu, Expanding eCall from cars to other means of transport, Journal of Information Systems & Operations Management, Vol. 10 No.2 / December, pag. 354-363
- [5] Tomasz Kaminski, Monika Ucinska, Ewa Kamiska- Effect analysis on the implementation of automatic emergency call system eCall- Journal of KONES Powertrain and Transport, Vol. 18, No. 4 2011
- [6] Pierpaolo Tona- EU-wide interoperable harmonised eCall- EU Emergency Services Workshop 2012 Riga 19/04/2012
- [7] Jerome Paris- eCall deployment in Europe- HeERO International Conference 15th November 2012 Zagreb
- [8] Botezatu Cezar, Bârcă Claudiu. Intelligent vehicle safety systems-eCall, Journal of Information Systems & Operations Management No. 2, December 2008, pag. 487-494, ISSN 1843-4711
- [9] Bârcă Claudiu Dan, Rareș Ropot, Sorin Dumitrescu –eCall Minimum Set Of Data (MSD)- Journal of Information Systems & Operations Management, 2009, vol. 3, issue 2, pages 428-439
- [10] Marco Annoni -eCall Deployment- HeERO International Conference 15th November 2012 Zagreb
- [1 1] Emilio Davila -European Commission DG Information Society. Unit ICT for Transport-Status of the eCall initiative- 6th EeIP Meeting Brussels, 24/3/11
- [12] Microsoft Developer Network,
- [13] Microsoft.Net Framework 3.5 Service Pack 1



Claudiu Dan BÂRCA graduated from Faculty of Computer Science for Business Management, Romanian American University in 2007, and holds a master degree in Economic Informatics since 2008 and a PhD in the field of Engineering Sciences since 2013. He is an assistant lecturer within the Faculty of Computer Science for Business Management having nine years of teaching experience. He also has good research and publishing activity: he was a member of the research teams of international and national projects. His core competences are in software programming and connected areas.