# Operational time predictor implementation on Symbian platform

Authors:

- Krinkin Mikhail
- Savelyev Dan
- Teslya Nikolay SPbETU, Russia

# Main goalIndustry makes:User wants to:

- More effective power
  Use lots of application
- Possible the use of different greedy features in mobile device.
- Use lots of applications at the same time.
- Be always on-line.

#### Main goal is to:

Reduce the power consumption of the mobile device.

### Sub task.

#### The small part in the global task is to:

- Predict precisely the operational time of the mobile device.
- Give to any user the possibility to use the mobile device as long as he wants, but has a limited functionality, that is the limited list of accessible applications.

#### **Basic terms**

<u>Application Profile</u> – a set of parameters (average: duration, frequency, current)

<u>User Profile</u> – a set of application profiles for each exact application.

**Event** – an event of one of the types:

- Application start or stop
- Change user profile
- New measurement

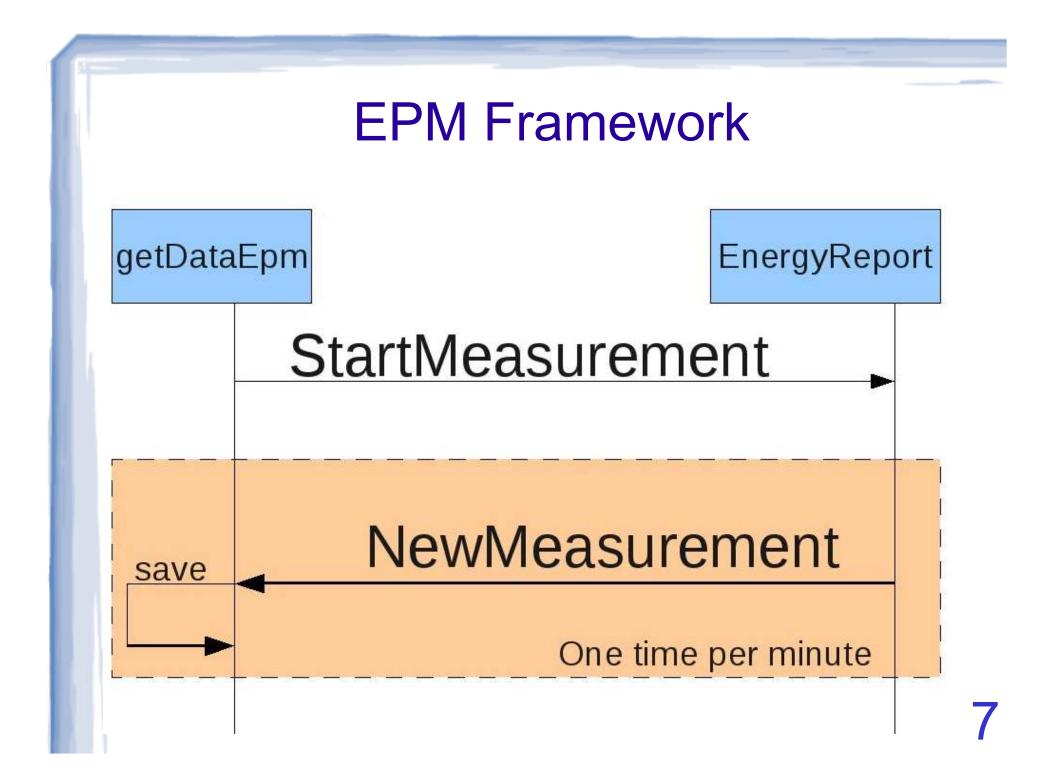
# **EPM logical structure**

Event	
+ type : EventType	EventType
+ GetEventType() : const EventType&	+type
+ GetTime() : quint32	
+ GetApplicationName(name : QString&) : bool	
+ GetProfileName(name : QString&) : bool	
+ GetMesure(current : quint32&, elapsedCharge : qint32&) : bool	
ApplicationProfile	PowerPredictor
+ GetName() : const QString&	+ PowerPredictor(parent : QObject*)
+ GetFrequence() : greal	+ ~ PowerPredictor()
+ GetDuration() : quint32	+ GetDeathTime() : gint32
+ GetCurrent() : quint32	+ GetCurrentUserProfile() : const UserProfile&
+ isCorrected() : bool	+ OnEvent(event : const Event&)
+ GetSpentCharge(time : const quint32) : qint32	
<u>Λ</u>	-currentProfile
	Sur di la Pono
UserProfile	
+ UserProfile(profile : const QString&)	
+ ReadFromFile() : bool	
+ WriteToFile() : bool	
+ isExists() : bool	
+ GetSpentCharge(time : const quint32) : qint32	
+ isContains(app : const ApplicationProfile&) : bool	
+ AddApplicationProfile(app : const ApplicationProfile&)	
+ GetApplicationProfile(app : const ApplicationProfile&) : const ApplicationProfile&	

## **Power Predictor**

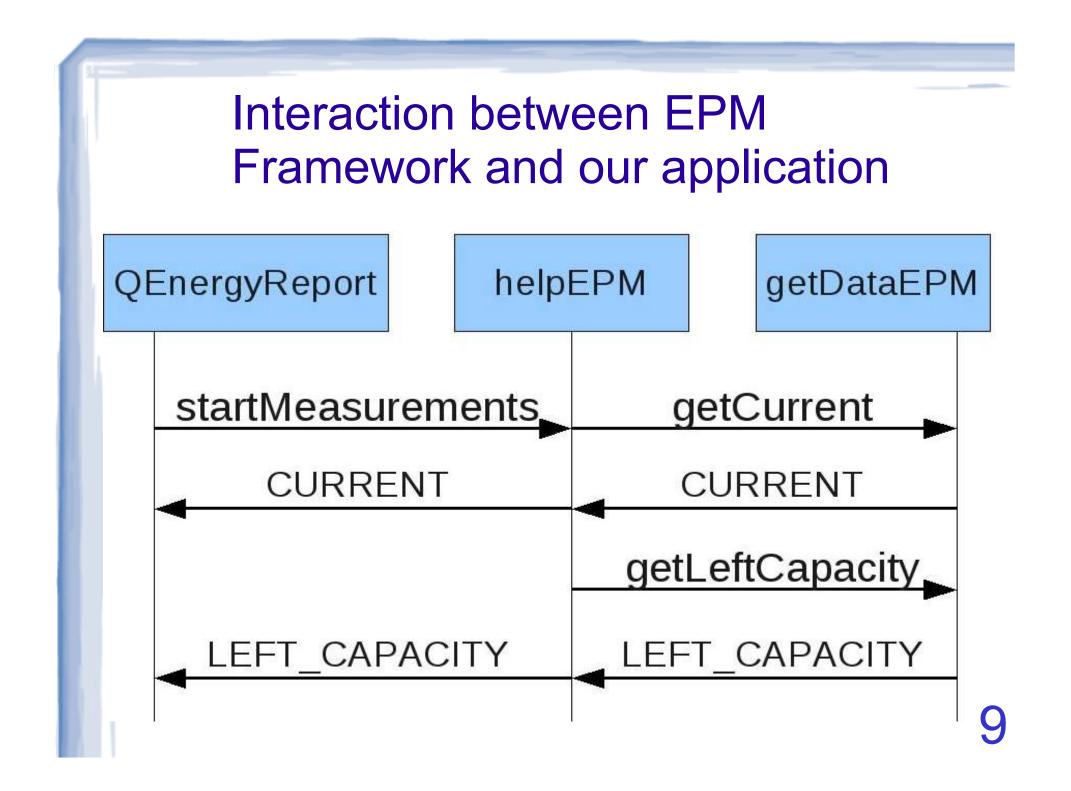
It is the main class of our application

- Its functions:
  - -React to events
  - -Emit signal "new death time"



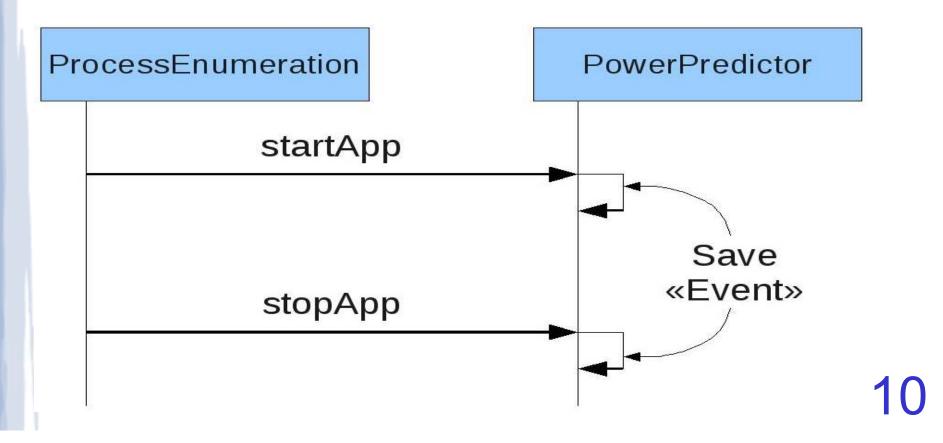
# **EPM Framework**

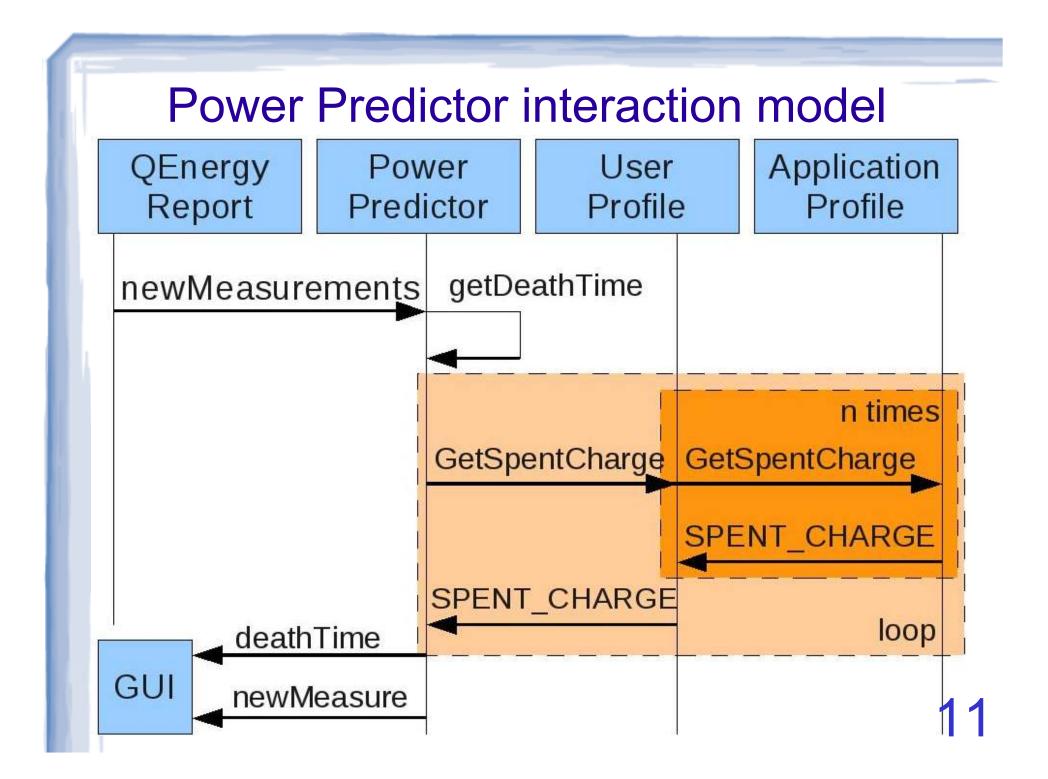
- Author: Gerard Bosch
- Provides specific parameters of the system:
  - -Voltage.
  - -Average power, spent during the last minute.
  - Left capacity.
  - Many others.

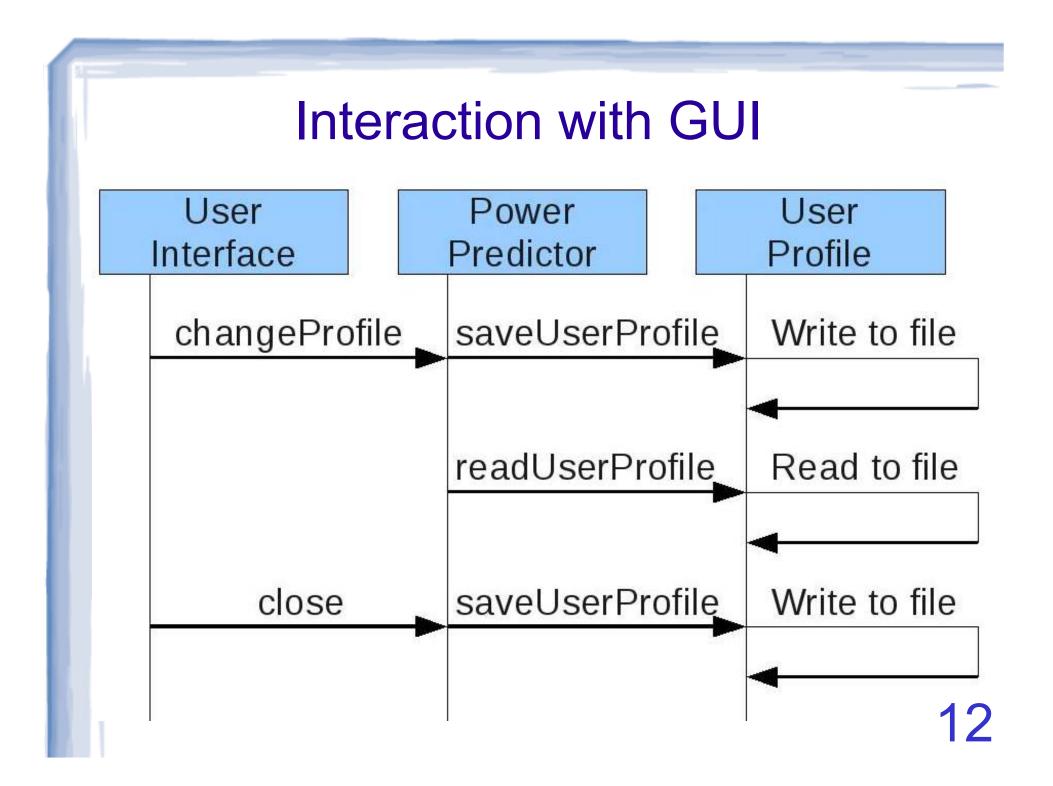


#### Interaction between process enumeration and power predictor

• Saving events to the list is necessary for calculation and changing the statistic data.







#### **User Interface**

