Autonomous Agent Model for Connectivity Management in M2M Communications

Ivaylo Atanasov[1], Anastas Nikolov[1] and Evelina Pencheva[1]

Abstract – **The large amount of diverse connected devices which interact with the environment without human intervention and the variety of Machine-to-Machine (M2M) applications is a real challenge for network operators. Embedding autonomic features in M2M system may reduce the operational cost. In this paper, we present a model of autonomous agent for connectivity management of M2M devices. The agent observes connectivity parameters of the device and based on operator policies determines the best bearer that has to be used by the device. Temporal logic is used to formalize the agent behaviour.**

Keywords – **Remote device management, Network bearer control, Autonomic behaviour, Temporal logic**

I. Introduction

The ubiquitous penetration of Machine-to-Machine (M2M) communications in different application area, the increased amount of equipment and the explosion of M2M services become business and technical challenges for network operators [1]. The reduction of network management complexity can be achieved by embedding autonomic features in operation support systems [2]. The autonomic system exposes reactive or proactive behavior based on external stimuli, following goals that are required to fulfill, policies, capabilities, principles of operation, experience and knowledge.

Currently, there is a lot of work conducted on autonomics by the research community. In [3], the authors discuss challenges and enablers that allow connected machines to evolve and act in a more autonomous way and propose architectural approach based on situational knowledge acquisition and analysis techniques in order to make machines aware of conditions and events affecting systems behavior. In [4], the authors propose a middleware architecture that connects the appropriate devices and applications, and is based on software agents representing devices and applications negotiating between each other on the terms by which the data can be used. In [5], the authors propose network architecture for remote monitoring and surveillance M2M networks with broadband satellite connection. In [6], it is proposed a flexible multi-agent approach, leveraging semantic-based resource discovery and orchestration for home and building automation applications. In [7], a generic architecture for multi-goal, adaptable and open autonomic systems, exemplified via the development of a concrete autonomic application for the smart micro-grid is proposed. While presenting high level architectural aspects of autonomous systems, these works discuss proprietary solutions and do not consider autonomics in generic M2M communications.

In this paper, we present a model of autonomous agent for connectivity management in M2M communications. The goal of the agent is to optimize the device connectivity parameters by choosing the best bearer that device should use, which is based on device location and device operator policy. First we describe the logic behind the autonomous connectivity management. Next, an overview on the knowledge base in the domain is presented. The conclusion summarizes the contribution.

II. M2M Connectivity Management Autonomous Agent

An agent is a thing that perceives from and acts on an M2M device in such way that the device goes through a sequence of states maximizing the performance measures. The problem in M2M self-optimization includes a goal and set of means to achieve the goal. One of the goals of M2M self-optimization is the usage of the best network bearer. An agent dedicated to M2M self-optimization (Connectivity Management Agent) reasons about and follows actions in order to achieve the goal. The process of reasoning what means it can do is called search. The Connectivity Management Agent is goal-based and solves the problem deciding what to do by finding sequences of actions that lead to the desirable operational state of M2M devices with cellular or wireless connectivity. Further we consider M2M devices with cellular or wireless connectivity. The agent actions can be viewed as transitions between M2M device states.

The problem solving of an M2M Connectivity Management Agent includes four stages: goal formulation, problem formulation, searching solution and execution. The Agent explores the current situation and draws the goal which helps to organize behavior by rejecting actions that result in a failure to achieve the operational state of the M2M device. The Agent draws the problem by deciding what transitions and states to consider following the operational state of M2M device. In general, an M2M Connectivity Management Agent faces with several options of possible sequences of actions because it does not know enough about the current device state. For example, there may be different reasons for device not answering (a connectivity problem, low battery level, a firmware failure, etc.). The Agent searches the solution space by examining different sequences of action. Once the solution is found, the agent carries out the identified actions in the execution stage.

The autonomous behavior of the proposed Connectivity Management Agent is based on the OMA Lightweight M2M

[1]The authors are the Faculty of Telecommunications at Technical University of Sofia, 8 Kl. Ohridski Blvd, Sofia 1000, Bulgaria, E-mails: iia@tu-sofia.bg’nikolov.anastas@gmail.com;enp@tu-sofia.bg

(LWM2M) [8]. LWM2M is a standard for a communication mechanism between M2M devices and M2M server platforms. The main motivation behind it is to reduce the degree of fragmentation in the area of M2M remote management.

The LWM2M defines a protocol between M2M devices in a role of clients and the server. The server is responsible for device registration and deregistration, firmware updates, device rebooting, monitoring connectivity parameters on the device, controlling bearers used by the device, configuration of APN and retrieval of device location data [9].

The Connectivity Management Agent in a role of LWM2M server is responsible for device registration, observation of device connectivity parameters and selection of best bearer for the device. The agent functionality for device registration is simplified due to space restrictions and to the facts that the focus is on bearer control.

We assume that the device operator has determined preferred bearers for both specific and normal areas. Each device supports traps, which means that the device is capable of monitoring the event and send notifications whenever it detects the event. The Connectivity Management Agent has to register for the capability in order to use it. OMA DiagMon Trap Events specification defines a number of standardized traps [10]. Geographic trap goes to active when a device enters into a specific geographic area. Whenever the device leaves that specific geographic area, the trap goes to inactive. Collection of measurement data (e.g. signal strength and quality) is important for the connectivity management when the device communicates over the wireless network. Received power trap can help the connectivity optimization process by triggering the Connectivity Management Agent when the received power of the device drops below the server-specific value. Whenever a device's received power drops below an agent-specified value (TrapActivePower), it causes this trap to go active. Alternatively, when device senses power rises above another agent-specified value (TrapInactivePower), it causes this trap to go inactive. In cases that the trap goes active or inactive, the device notifies the registered agent. The device can have several instances of this kind of trap to monitor various network types (e.g. WiFi, WCDMA, LTE etc). At any time the agent can query the device about its location and about the connectivity parameters, i.e. parameters include the used network bearer, available network bearers, signal strength as well as network identities.

Usually, the goal of Connectivity Management Agent is that the managed device to use the best networks bearer. One of the policies of choosing the best network bearer is as follows. If the device is in the specified area and the signal strength of the preferred bearer in area is higher than the specified value of TrapActivePower, then the best bearer is the preferred one for this area. When the device is out of the specified area and the signal strength of the preferred bearer out of area is higher than the specified value of TrapActivePower, then the best bearer is the preferred out of area one. If the signal strength of the preferred bearer is lower than the specified value of TrapActivePower, then the best bearer is the available bearer with highest signal strength.

The logic behind the Connectivity Management Agent behavior might be described as a temporal sequence. On successful device registration, the agent configures geographic traps and received power traps. The agent queries the device about its location and about connectivity parameters. Based on the location, the signal strength of the used network bearer and available bearers, and the best bearer policy the agent performs a bearer selection procedure for the device. After selecting the best bearer the device is in operational state. During this state, the device may send notifications about traps in case the respective event occurs and the agent performs the bearer selection procedure.

III. Knowledge Base Model for Device Connectivity Management

We use predicates to express the facts, to show the exchange of messages between the client and server, and to describe the device states as seen by the agent.

Excellent(*b*, *x*) becomes true when the received signal strength of bearer *b* by the device *x* is higher than the specified value of TrapInactivePower. Good(*b*, *x*) becomes true when the received signal strength of bearer *b* by the device *x* is between the specified values of TrapActivePower and TrapInactivePower. If the device senses signal strength of *b* below the specified value of TrapActivePower then the Bad(*b*, *x*) gets true. In case the device *x* uses bearer *b* then Used(*b*, *x*) is true. InArea(*a*, *x*) is true when the device *x* is in the area *a*. Predicates PreferredIn(*b*, *a*) and PreferredOut(*b*, *a*) are true when bearer *b* is preferred bearer in area *a* and out of area *a* respectively. The express the fact that bearer *b* is available for device *x* the Available(*b*, *x*) is used. When there are no available bearers for *x* except the used one then AvailableEmpty(*x*) is true. BadPreferred(*x*) is true the received signal strength of preferred bearer by the device *x* is bad. Best(*b*, *x*) is true if the received signal strength of *b* is the maximal one for device *x*. PowerTrapActive(*x*, *b*) gets true when the power trap goes active and the signal strength of used bearer *b* by device *x* becomes bad. PowerTrapInactive(*x*, *b*) gets true when the power trap goes inactive and the signal strength of used bearer *b* by device *x* becomes excellent.

The behavior of the Connectivity Management Agent is described by temporal logic. We use a minimal set of standard notations $\mathcal{G}$ for always, $\mathcal{U}$ for until, and $\mathcal{N}$ for next.

The agent considers the following statement when explores the current device state, formulates the problem, searches the solution and performs actions.

The device x is unregistered until a registration request is received:

$$\mathcal{G}(\text{Unregistered}(x) \rightarrow \top\, \mathcal{U}\, \text{reg}_{req}(x)) \quad (1)$$

After successful device registration, the server configures geo trap and power trap.

If the device *x* is unregistered and a registration request is received then a registration response is sent, and a request for geo trap configuration is sent, and the state becomes WaitGeoAck:

$$\mathcal{G}(\text{Unregistered}(x) \wedge \text{reg}_{req}(x) \rightarrow \text{reg}_{res}(x) \wedge \neg\text{BadPreferred}(x) \wedge \text{configGeoTrap}_{res}(x) \wedge \mathcal{N}\text{WaitGeoAck}(x)) \quad (2)$$

The device x is in WaitGeoAck state until the server receives a response of geo trap configuration:

$$\mathcal{G}(\text{WaitGeoAck}(x)\rightarrow\top\,\mathcal{U}\,\text{configGeoTrap}_{res}(x)) \quad (3)$$

If the state is WaitGeoAck and a response of geo trap configuration is received then a request for configuration of power trap is sent and the state becomes WaitPowerAck:

$$\mathcal{G}(\text{WaitGeoAck}(x)\wedge\text{configGeoTrap}_{res}(x)\rightarrow \text{configPowerTrap}_{req}(\text{x})\wedge\,\mathcal{N}\text{WaitPowerAck}(x)) \quad (4)$$

The device x is in WaitPowerAck state until the server receives a response of power trap configuration:

$$\mathcal{G}\,(\text{WaitPowerAck}(x)\rightarrow\top\,\mathcal{U}\,\text{configPowerTrap}_{res}(x)) \quad (5)$$

After successful configuration of geo and power traps the server requests the device location and the device connectivity parameters.

$$\mathcal{G}(\text{WaitPowerAck}(x)\wedge\text{configPowerTrap}_{res}(x)\rightarrow \text{getLocation}_{req}(\text{x})\wedge\mathcal{N}\text{WaitLocation}(\text{x})) \quad (6)$$

The device x is in WaitLocation state until a location response is received.

$$\mathcal{G}(\text{WaitLocation}(x)\rightarrow\top\,\mathcal{U}\,\text{getLocation}_{res}(x)) \quad (7)$$

The location response will allow the server to determine whether the device x is in are a:

$$\mathcal{G}(\text{WaitLocation}(x)\wedge\text{getLocation}_{res}(x)\rightarrow \text{connParameters}_{req}(x)\wedge\mathcal{N}\,\text{WaitConnectivity}(x)) \quad (8)$$

The device x is in WaitConnectivity state until a connectivity parameters response is received. The connectivity parameters response will contain the signal strength of used bearer b by device x and available bearers for device x.

$$\mathcal{G}(\text{WaitConnectivity}(x)\rightarrow\top\,\mathcal{U}\text{connParameters}_{res}(x))(9)$$

Equations from (10) to (13) refer to bearer selection procedure when the device x is in area a.

When the used bearer of x is b, and b is the preferred bearer in area a, and the signal strength of b is excellent or good, then the state becomes Operational:

$$\mathcal{G}(\text{WaitConnectivity}(x)\wedge\text{connParameters}_{res}(x)\wedge \text{InArea}(a, x)\wedge\text{PreferredIn}(b, a)\wedge\text{Used}(b, x)\wedge (\text{Excellent}(b, x)\vee\text{Good}(b, x))\rightarrow\,\mathcal{N}\text{Operational}(x))(10)$$

When the used bearer b of x is the preferred one, and the signal strength of b is bad, and c is available bearer for device x and c is the best bearer then a request to select bearer c is sent, and the state becomes WaitBearerAck:

$$\mathcal{G}(\text{WaitConnectivity}(x)\wedge\text{connParameters}_{res}(x)\wedge \text{InArea}(a,x)\wedge\text{PreferredIn}(b,a)\wedge\text{Used}(b,x)\wedge\text{Bad}(b,x)\wedge \text{Available}(c, x)\wedge\text{Best}(c, a)\rightarrow\text{select}_{req}(x,c)\wedge \text{BadPreferred}(x)\wedge\mathcal{N}\text{WaitBearerAck}(x)) \quad (11)$$

The bearer c selection procedure takes place when c is available and preferred bearer, and the received signal strength of c by device x is not bad:

$$\mathcal{G}(\text{WaitConnectivity}(x)\wedge\text{connParameters}_{res}(x)\wedge \text{InArea}(a,x)\wedge\text{Used}(b,x)\wedge\neg\text{PreferredIn}(b,a)\wedge \text{Available}(c,x)\wedge\text{PreferredIn}(c, a)\wedge\neg\text{BadPreferred}(x) \rightarrow\text{select}_{req}(x, c)\wedge\mathcal{N}\text{WaitBearerAck}(x)) \quad (12)$$

When the used bearer of x is not the preferred one, and c is available and preferred bearer in area a, and the received signal strength of c by device x is bad, and d is the best available bearer then the server initiates bearer d selection procedure:

$$\mathcal{G}(\text{WaitConnectivity}(x)\wedge\text{connParameters}_{res}(x)\wedge\text{InArea}(a, x)\wedge \text{Used}(b,x)\wedge\neg\text{PreferredIn}(b,a)\wedge\text{Available}(c,x)\wedge \text{PreferredIn}(c,a)\wedge\text{BadPreferred}(x)\wedge\text{Best}(d,a)\wedge \text{Available}(d,\text{x})\rightarrow\text{select}_{req}(x,d)\wedge \mathcal{N}\text{WaitBearerAck}(x)) \quad (13)$$

Equations from (14) to (17) refer to bearer selection procedure when the device x is out of area a.

When the used bearer b of x is the preferred one, and the signal strength of b is excellent or good, then the state becomes Operational:

$$\mathcal{G}(\text{WaitConnectivity}(x)\wedge\text{connParameters}_{res}(x)\wedge \neg\text{InArea}(a, x)\wedge\text{PreferredOut}(b, a)\wedge\text{Used}(b, x)\wedge (\text{Excellent}(b, x)\vee\text{Good}(\text{b, x}))\rightarrow\mathcal{N}\text{Operational}(x)) \quad (14)$$

When the used bearer b of x is the preferred one, and the signal strength of b is bad, and c is the best available bearer then a bearer c selection procedure take place:

$$\mathcal{G}\,(\text{WaitConnectivity}(x)\wedge\text{connParameters}_{res}(x)\wedge \neg\text{InArea}(a,x)\wedge\text{PreferredOut}(b,a)\wedge\text{Used}(b,x)\wedge\text{Bad}(b,x)\wedge \text{Available}(c,x)\wedge\text{Best}(c,a)\rightarrow\text{select}_{req}(x,c)\wedge \text{BadPreferred}(x)\wedge\mathcal{N}\text{WaitBearerAck}(x)) \quad (15)$$

In case the used bearer b of x is not the preferred one, and c is available preferred bearer, and the received signal strength of c by device x is not bad, then the server request selection of bearer c:

$$\mathcal{G}(\text{WaitConnectivity}(x)\wedge\text{connParameters}_{res}(x)\wedge \neg\text{InArea}(a, x)\wedge\text{Used}(b,x)\wedge\neg\text{PreferredOut}(b,a)\wedge \text{Available}(c, x)\wedge\text{PreferredIn}(c, a)\wedge\neg\text{BadPreferred}(x) \rightarrow\text{select}_{req}(x,c)\wedge\,\mathcal{N}\text{WaitBearerAck}(x)) \quad (16)$$

When the signal strength for the preferred bearer c is bad and d is the best available bearer then the server request selection of bearer d:

$$\mathcal{G}(\text{WaitConnectivity}(x)\wedge\text{connParameters}_{res}(x)\wedge \neg\text{InArea}(a,x)\wedge\text{Used}(b,x)\wedge\neg\text{PreferredOut}(b,a)\wedge \text{Available}(c,x)\wedge\text{PreferredIn}(c, a)\wedge \text{BadPreferred}(x)\wedge\text{Best}(d,\text{a})\wedge\text{Available}(d,x)\rightarrow \text{select}_{req}(x, d)\wedge\mathcal{N}\text{WaitBearerAck}(x)) \quad (17)$$

When there are no available bearers for device x, and the received signal strength of b by device x is good the state becomes Operational:

$$\mathcal{G}(\text{WaitConnectivity}(x)\wedge\text{connParameters}_{res}(x)\wedge \text{Used}(b,x)\wedge(\text{Excellent}(b, x)\vee\text{Good}(b, x))\wedge \text{AvailableEmpty}(x)\rightarrow\,\mathcal{N}\,\text{Operational}(x)) \quad (18)$$

The server considers the device x unregistered when the received signal strength of the used bearer b is bad and there are no available bearers:

$$\mathcal{G}(\text{WaitConnectivity}(x)\wedge\text{connParameters}_{res}(x)\wedge \text{Used}(b,x)\wedge\text{Bad}(b,x)\wedge\text{AvailableEmpty}(x)\rightarrow \mathcal{N}\text{Unregistered}(x)) \quad (19)$$

The device x is in WaitBearerAck state until the server receives a response of bearer selection procedure:

$$\mathcal{G}(\text{WaitBearerAck}(x)\rightarrow\top\,\mathcal{U}\,\text{selectBearer}_{req}(x, b))(20)$$

When a beater selection response is received, the server waits for device re-registration:

$$\mathcal{G}(\text{WaitBearerAck}(x) \wedge \text{selectBearers}_{res}(x) \rightarrow \mathcal{N}\text{WaitReregistration}(x)) \quad (21)$$

The device x is in WaitReregistration state until the server receives a registration request:

$$\mathcal{G}(\text{WaitReregistration}(x) \rightarrow \top\, \mathcal{U}\, \text{reg}_{req}(x)) \quad (22)$$

Upon successful device re-registration the server request device location:

$$\mathcal{G}(\text{WaitReregistration}(x) \wedge \text{reg}_{req}(x) \rightarrow \text{reg}_{res}(x) \wedge \neg\text{BadPreferred}(x) \wedge \text{getLocation}_{req}(x) \wedge \mathcal{N}\text{WaitLocation}(x)) \quad (23)$$

The device x is in Operational state until the server receives a notification about signal strength change of used bearer by x or a notification about change of x location:

$$\mathcal{G}(\text{Operational}(x) \rightarrow \top\, \mathcal{U}\, (\text{notifyPower}_{req}(x) \vee \text{notifyGeo}_{req}(x))) \quad (24)$$

In Operational state, when a notification about location change of device x is received, the server sends a response of geo trap notification and requests device connectivity parameters:

$$\mathcal{G}(\text{Operational}(x) \wedge \text{notifyGeo}_{req}(x) \rightarrow \text{notifyGeo}_{res}(x) \wedge \text{connParameters}_{req}(x) \wedge \mathcal{N}\text{WaitConnectivity}(x)) \quad (25)$$

When the power trap becomes inactive in Operational state, the server sends a response of power trap notification and the state remains Operational:

$$\mathcal{G}(\text{Operational}(x) \wedge \text{notifyPower}_{req}(x,b) \wedge \text{PowerTrapInactive}(x) \rightarrow \text{notifyPower}_{res}(x,b) \wedge \mathcal{N}\text{Operational}(x)) \quad (26)$$

Activation of power trap in Operational state means that the signal strength becomes bad and the server sets the margin timer:

$$\mathcal{G}(\text{Operational}(x) \wedge \text{notifyPower}_{req}(x,b) \wedge \text{PowerTrapActive}(x) \rightarrow \text{notifyPower}_{res}(x,b) \wedge \text{setTmargin}(x) \wedge \mathcal{N}\text{WaitMargin}(x)) \quad (27)$$

The device x is in WaitMargin state until the server receives a notification about signal strength change of used bearer by x or a notification about change of x location:

$$\mathcal{G}(\text{WaitMargin}(x) \rightarrow \top\, \mathcal{U}\, (\text{notifyPower}_{req}(x,b) \vee \text{notifyGeo}_{req}(x) \vee \text{Tmargin}(x))) \quad (28)$$

Notification that the power trap is inactive in WaitMargin state means that the signal strength becomes excellent and the server and resets the margin timer:

$$\mathcal{G}(\text{WaitMargin}(x) \wedge \text{notifyPower}_{req}(x,b) \wedge \text{PowerTrapInactive}(x) \rightarrow \text{notifyPower}_{res}(x,b) \wedge \text{resetTmargin}(x) \wedge \mathcal{N}\,\text{Operational}(x)) \quad (29)$$

In WaitMargin, when the margin timer expires, the server sends requests device connectivity parameters:

$$\mathcal{G}(\text{WaitMargin}(x) \wedge \text{Tmargin}(x) \rightarrow \text{connParameters}_{req}(x) \wedge \mathcal{N}\,\text{WaitConnectivity}(x)) \quad (30)$$

The device may change its location while it is in WaitMargin state:

$$\mathcal{G}(\text{WaitMargin}(x) \wedge \text{notifyGeo}_{req}(x) \rightarrow \text{notifyGeo}_{res}(x) \wedge \mathcal{N}\,\text{WaitMargin}(x)) \quad (31)$$

When the device is unregistered due to connectivity problems, it may wait for some time and try to register again.

For brevity sake the behavioral model of the Connectivity Management Agent is given without parts like re-registration which is caused by registration timer expiry, APN activation due to change of network bearer etc.

IV. Conclusion

Embedding autonomic features in M2M systems reduces operational costs. Following the goal of ensuring a reliable and efficient M2M network, autonomous agents that perceive from the M2M ecosystem and act upon it can contribute to self-optimization.

We present an algorithm for choosing the best bearer and model the behavior of Connectivity Management Agent which selects radio technologies in wireless environment according to current radio conditions. The agent monitors the connectivity parameters on M2M devices and based on the device location and policies defined by the device operator, selects the best network bearer to be used by the device. The model is formally described by temporal logic. Our future work will include more details related to APN configuration, device rebooting and firmware update.

References

[1] P. Watson, K. C. Gupta, “EM-ANN Models for Microstrip Vias and Interconnects”, IEEE Trans., Microwave Theory Tech., vol. 44, no. 12, pp. 2395-2503, 1996.

C. Pereira, A. Aguiar, “Towards Efficient Mobile M2M Communications: Survey and Open Challenges,” Sensors 2014, no. 14, 19582-19608; doi:10.3390/s141019582, pp.19582-19608.

[2] J. Sachs, N. Beijar, P. Elmdahl, J. Melen, F. Militano, P. Salmela, “Capillary networks – a smart way to get things connected,” Ericsson Review, 2014, no. 8, pp. 2-8.

[3] D. Kyriazisa, T. Varvarigoua, “Smart, autonomous and reliable Internet of Things,” International Workshop on Communications and Sensor Networks (ComSense’2013), International Workshop on Communications and Sensor Networks, ComSense’2013, Procedia Computer Science, 2013, pp. 442 – 448.

[4] S. Vassaki, G. Pitsiladis, C. Kourogiorgas, M. Poulakis, A. Panagopoulos, G. Gardikis, S. Costicoglou, “Satellite-based sensor networks: M2M sensor communications and connectivity analysis”, International Conference on Telecommunications and Multimedia (TEMU), Greece, 2014, pp.132–137.

[5] K. Misura, M. Zagar, “Internet of things cloud mediator platform,” International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO), 2014, pp.1052-1056

[6] M. Ruta, F. Scioscia, G. Loseto, E. Di Sciascio, “Semantic-Based Resource Discovery and Orchestration in Home and Building Automation: A Multi-Agent Approach,” IEEE Transactions on Industrial Informatics, vol. 10, no. 1, 2014, pp.730-741.

[7] S. Frey, A. Diaconescu, D. Menga1, I. Demeure, “Towards a generic architecture and methodology for multi-goal, highly-distributed and dynamic autonomic systems,” International Conference on Autonomic Computing (ICAC), 2013, pp.201-212.

[8] G. Klas, F. Rodermund, Z. Shelby, A. Akhouri, J. Holler, “Lightweight M2M”: Enabling Device Management and Applications for the Internet of Things,” white paper, 2014.

[9] Open Mobile Alliance, Enabler Test Specification for Lightweight M2M, 2015, OMA-ETS-LightweightM2M-V1_0-20150203-C.

[10] Open Mobile Alliance, Diagnostics and Monitoring Trap Events Specifications,2013,OMA-TS-DiagMonTrapEvents-V1_2-20131008-A