# RFC - Events system

Metacosm Development Team

## Revision History

| | | |
|---|---|---|
| Revision 0.4 | May, 7th 2000 | Revised by: Ruffy |
| More detailed explanations - examples added | | |
| Revision 0.3 | May, 3th 2000 | Revised by: Ruffy |
| Major changes both in the Event concept and in Events system - examples added | | |
| Revision 0.2 | March, 29th 2000 | Revised by: Ruffy |
| Conversion to metacosm-DTD - Events types | | |
| Revision 0.1 | March, 5th 2000 | Revised by: Ruffy |
| First written version | | |
| Revision First draft and discussions | February, 19th 2000 | Revised by: Elkine, Horus, Janselmeer, Ruffy |

The goal of this document is to define the use of events and the concept of events system, in the Metacosm project.

# 1. License

# 2. Metacosm

For more information on Metacosm, check our site: Metacosm (http://metacosm.org).

# 3. Notations

Some terms are defined in the Vocabulary-RFC, which is available on the official web site. Their first letter is in uppercase.

# 4. Event

### Event

An Event is something happening somewhere, which can be important for somebody.

something: many different things are Events and Events have many different types.

happening: an Event is caused by an Action, triggered by a timer or randomly produced. It can be created by the Game, by a Bot, by a Player or by something external.

somewhere: it can be precisely located (in a Place or in a World) or not located (generated by the Game).

important: an Event is something susceptible to make somebody or something have a reaction. Important means of some value, but not necessary for everybody.

somebody: somebody (or something) will react to the Event. It can be one person, several people, everybody, even nobody ('which *can* be'): the Game, a part of the Game, parts of the Game, a Player, several Players, all Players, a Bot, several Bots, all Bots, nobody/nothing or a mix of these possibilities.

> Perceptions system transports low level information. Players can convert these low level information into high level information by themselves, because they are able to think and deduce. We aren't able to make bots really think. So we introduce the Events system which transports high level information: bots can react to Events. Players can also take part of the info contained in Events (but they obviously are free to react or not).

An Event contains only data (not the references on data): real info can be modified after an Event happen, and somebody who knows about the Event doesn't necessarily know about the change. Moreover an Event can contain false/wrong data (transmission error, lie, etc).

# 5. Events system

Events can be produced (thrown) by several ways, as we said in the Event section.

The "patient zero" could learn the info

- directly (rare case)

**Example 1. God of War**

> The God of War is unhappy and asks for more blood. All his believers will be directly notified (because he's an omnipotent god).

- as a direct witness (transported with a Stimulus)

**Example 2. The robbery**

> Raoul is attacked by a robber. The clients of the bakery and he know a robbery happened.

**Example 3. The king**

The king makes a speech and announces new trade agreements with an other kingdom. All the crowd knows about this event.

- could be the source of the Event

**Example 4. Lier**

Sedecias launch false rumour on Raoul: the baker is a murderer. The false Event'll propagate.

**Example 5. Pyromaniac**

The pyromaniac knows that he burns the forest.

There are two ways to diffuse them:

- through Items: Items can be used to transport information (tape recorder, letter, CD, marble tablet, book, computer, ...)
- through Creatures: Creatures can talk, gossip, lie, etc.

## 5.1. Direct transmission

The direct transmission (Controllers are directly noticed) relies on EventsDispatchers.

Events are transmitted only to Entities which ask for them (so which can interpret them).

Some Entities (like gods) are allowed to produce an Event and send it to an EventsDispatcher. Everybody can receive Events by registering for some Events type to an EventsDispatcher (the faithful).

This way implies nothing about perception.

> Events reception should be logical or at least plausible... So somebody should have good reasons to listen to Events.

## 5.2. Transmission via perceptions

The source of the Event'll send Stimuli to the StimuliDispatcher (and to the EventsDispatcher). The Event is sent linked with these Stimuli.

The StimuliDispatcher'll propagate these Stimuli+Event. As for Stimuli, Events can be not understood or misunderstood. The receiver can reemit what it perceived (can be used for rumours, reports, info diffusion, etc).

A Event can be linked to zero, one or more Stimuli. A Stimulus can be linked to zero, one or more Events.

**Example 6. Dead man**

"You see a dead man."

A 'death' Event is linked to the visual Stimulus, so the Entity can 'manage' this death and reacts logically (burying, stealing, doing nothing, spit on its enemy corpse, cry on its mother corpse, call the police, start investigation, etc.

**Example 7. The sleeping man**

You see a sleeping man and you make a mistake.

A wrong ’death’ Event is linked to the visual Stimulus. You’ll probably be surprised if you try to steal the man. If you go to the police, you’ll repeat what you see (Stimulus + Event)

## 5.3. Transmission via Items

The Event is linked to some Item (a media in fact). If an Entity is able to read the content of the media, he/she/it can learn the Event.

To have the Item is not enough: few people can read a CD-ROM just by touching it and sensing the holes on its surface...

Moreover content can be ciphered or incomprehensible for the Entity (unknown language, unknown domain, etc).

So Creatures can exchanged info by writing letters, by calling, etc.

## 5.4. Transmission via Creatures

Cf Memory-RFC for rumours mechanism

## 5.5. EventsDispatcher

The EventsDispatcher has two roles: to log all Events and to relay some of these Events.

All Events are sent to the EventsDispatcher (and some to a StimuliDispatcher).

Some Entities could subscribe to an EventsDispatcher. Enrolment is a way to say "I want to be informed of Events types A, B and C." You don’t declare what you plan to generate but only what you wish to receive.

# 6. More examples

**Example 8. The pyromaniac's return**

GrosBill burns a forest.

1. Goddess of Nature sends clerics (with maces) to teach him he's a bad bad guy. (twice direct transmission twice)
2. Newspapers talks about. (witnesses report to some journalists)
3. King sends guards to catch him. (witnesses report to the King + mission)
4. Ecosystem is modified, weather is modified. (the local Place receives the Event and modifies the weather)

**Example 9. A robbery again**

A robbery happens in a temple, so an Event is thrown.

- Bad example: Event is received by guards, so they go to the temple. How can we explain they know such an info?
- Good example: Event is received by the temple goddess, so she can cast a malediction on guilties.
- Another good example: in fact, robbers triggered a magic alarm. A bipper sounds at the guard post, so guards go to the temple.

**Example 10. I'm a god**

For a player: You see a marvellous woman, dressed in white. She's incredibly beautiful and can't be at all a simple woman. She's too perfect. And moreover, there is a golden aura around her.

For a bot: ’divine presence’ Event

**Example 11. Long live the King!**

For a player: You read the newspaper and discover during the night the King is dead.

For a bot: ’famous person death news’ Event

# 7. Why Events?

Advantages are flexibility, extensibility, its commonly use in very famous projects (X11, window managers, etc). It’s quite simple too.

Drawback is overload for the Game, but we need these features...

For us, it has some other advantages, such as providing a AI management for Bots (to simulate perceptions understanding and reactions) and a dynamic game (by managing some laws and some actions/reactions).

# 8. Events vs Stimuli?

Read the Perceptions-RFC, for more info on Stimuli and Perception system.

Stimulus concept is more Player-oriented, and Event concept is more Bot-oriented. It’s really easy for a Player to interpret Stimuli, but it will be really hard to do complete Stimuli recognition for Bots. It would be far too intricated to try to make the bots realize there is an aggression when they receive a scream stimulus for example: there could be an aggression without a single scream, and there could be a scream without an aggression. To determine that, the bot would have to process a lot of other stimuli which would perhaps not be available. Each time, it would be like trying to put together the pieces of a puzzle, and that would be too difficult and too long. Events make the situation much easier to handle. That’s why many Events are more specifically for Bots (they can be used indirectly for Players too).

# 9. Events types

This list is NOT exhaustive. It will be complete as soon as we have new strange ideas to put in.

We have chosen a hierarchical order. If something can be classified in several branches, several Events will be produced.

We choose not to detail too much: such a list is part of the game instance, not of the framework. We won’t detail PlayerEvents into PlayerEntranceEvent, PlayerLeavingEvent, PlayerDeletionEvent, PlayerBanningEvent, PlayerIdleEvent, etc. Some instances can, for example, propose a chat between Players and use Events to manage it.

## 9.1. WeatherEvents

WeatherEvents represent changes in the weather. They are local to one Place.

- TemperatureEvents (increase, decrease, etc)
- PrecipitationEvents (dryness, normal, etc)
- PressureEvents (low, normal, high, etc)

## 9.2. TimeEvents

TimeEvents represent the course of time. They are local to one World. They are thrown by the Game timer.

- DayEvents (today is a next day)

- MonthEvents (today begins a new month)
- YearEvents (today begins a new year)
- SeasonEvents (today begins a new season)
- TideEvents (rising/ebb/low/high tide)

## 9.3. CosmologicalEvents

CosmologicalEvents represent changes in the World that are due to cosmos.

- SunEvents (sunset, zenith, sunrise)
- SatelliteEvents (moonset, moonrise)
- StarEvents (evolution far in the sky)

## 9.4. EcologicalEvents

EcologicalEvents represent major changes in the Ecology domain.

- DisasterEvents (pollution, flood, tsunami, etc)
- SpeciesEvents (species birth, species death)
- PlaceEvents (major changes in a Place (desert became plain))

## 9.5. PoliticalEvents

PoliticalEvents regroup changes in politics.

- FunctionEvents (elections, putsches, death, etc)
- RelationEvents (treaties, negotiations, etc)

## 9.6. EcosystemEvents

EcosystemEvents represent changes in a local ecosystem.

- FloraEvents (growing, etc)
- FaunaEvents (Population increase, decrease, etc)

## 9.7. ReligiousEvents

Events linked to Deities or Cults.

- DeityEvents (divine intervention, divine appearance, etc)
- CultEvents (celebrations, etc)

## 9.8. MagicalEvents

Events linked to magic

## 9.9. JusticeEvents

Events linked to justice

- IllegalActionEvents (to steal, to rape, to kill, etc)
- JusticeCourtEvents (judgements, etc)

## 9.10. MilitaryEvents

Events linked to armies (wars, fights, etc)

## 9.11. EconomicalEvents

Events linked to Economy (Stock Exchange evolution, products prices evolution, etc)

## 9.12. CreatureEvents

Events linked to Creatures (birthday, death, etc)

## 9.13. PlaceEvents

Events linked to activity in a Place (building, etc)

## 9.14. ItemEvents

Events linked to Items