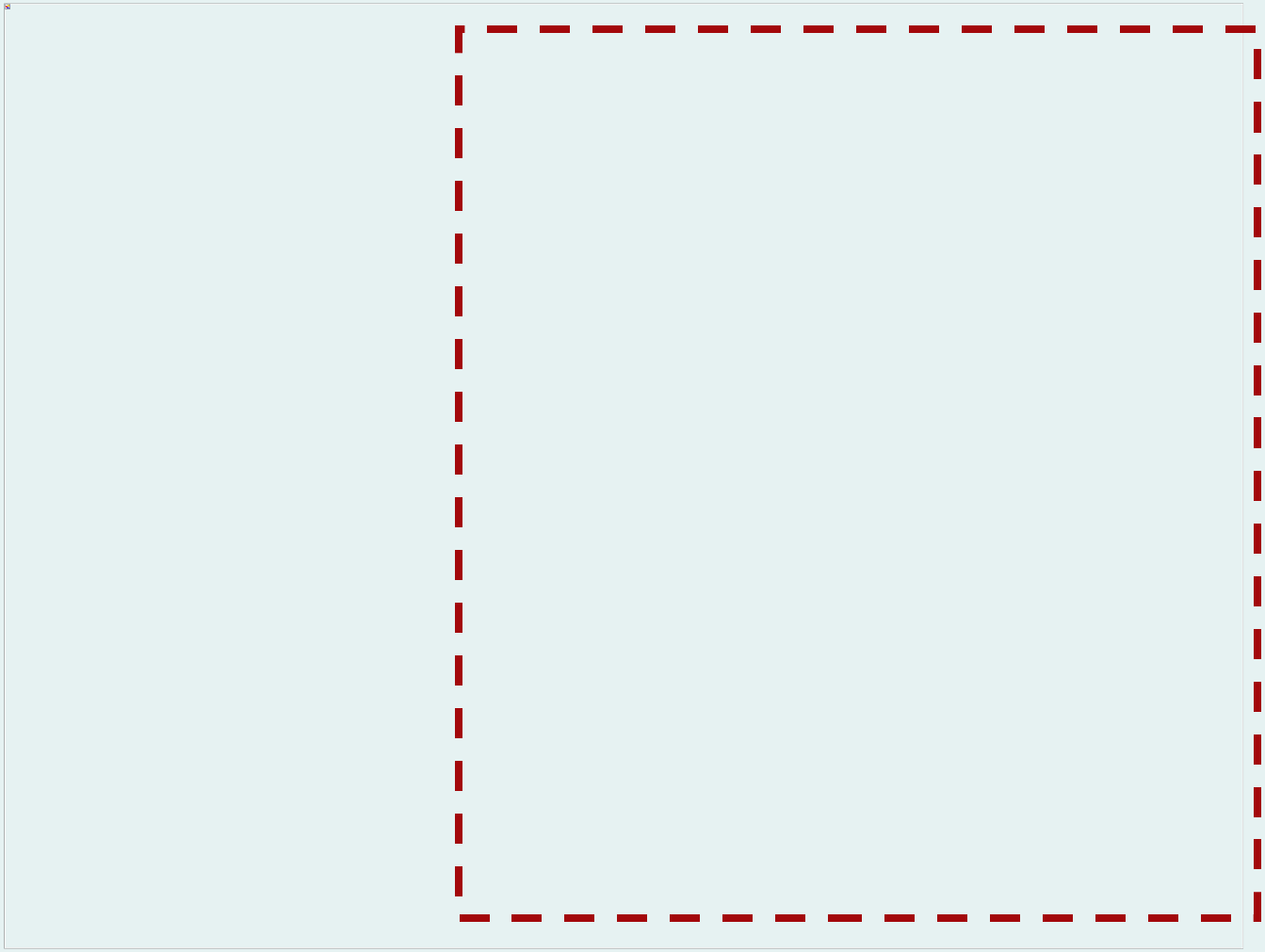
Reasoning Module Design

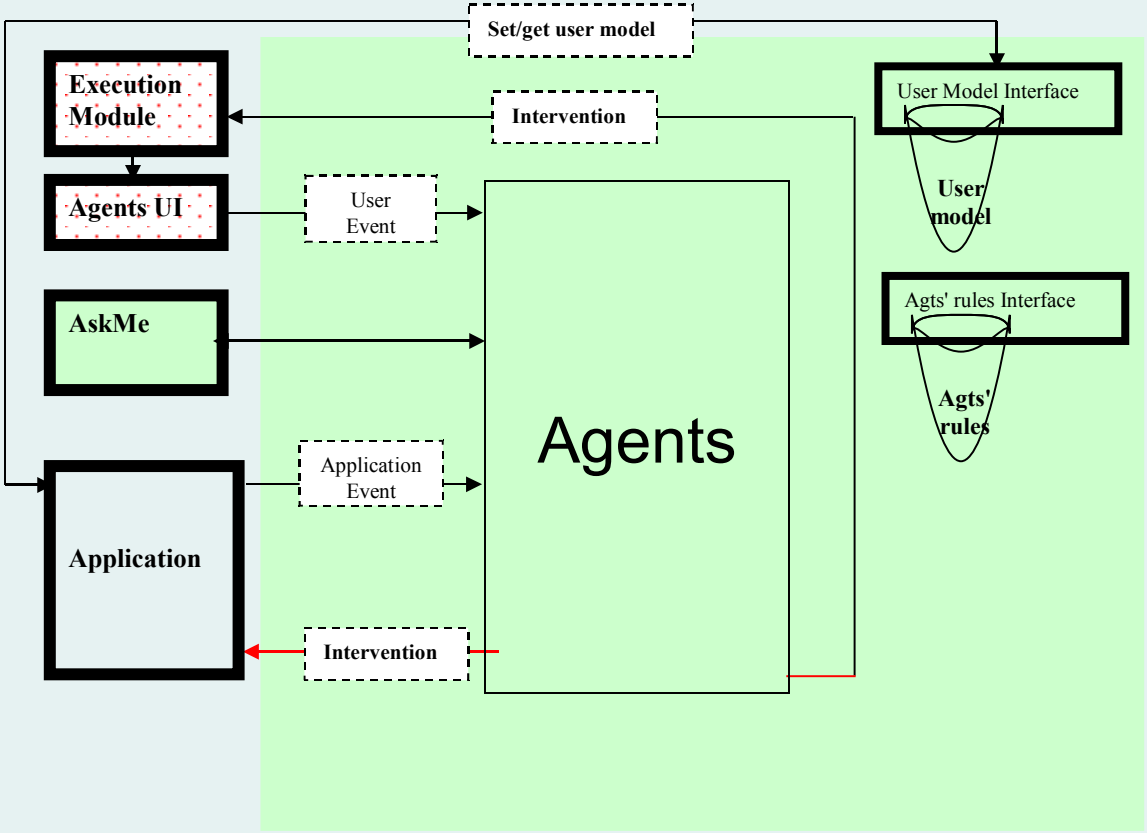
C. Roda, D. Clauzel, M. Raglianti
American University Of Paris

Atgentive Design

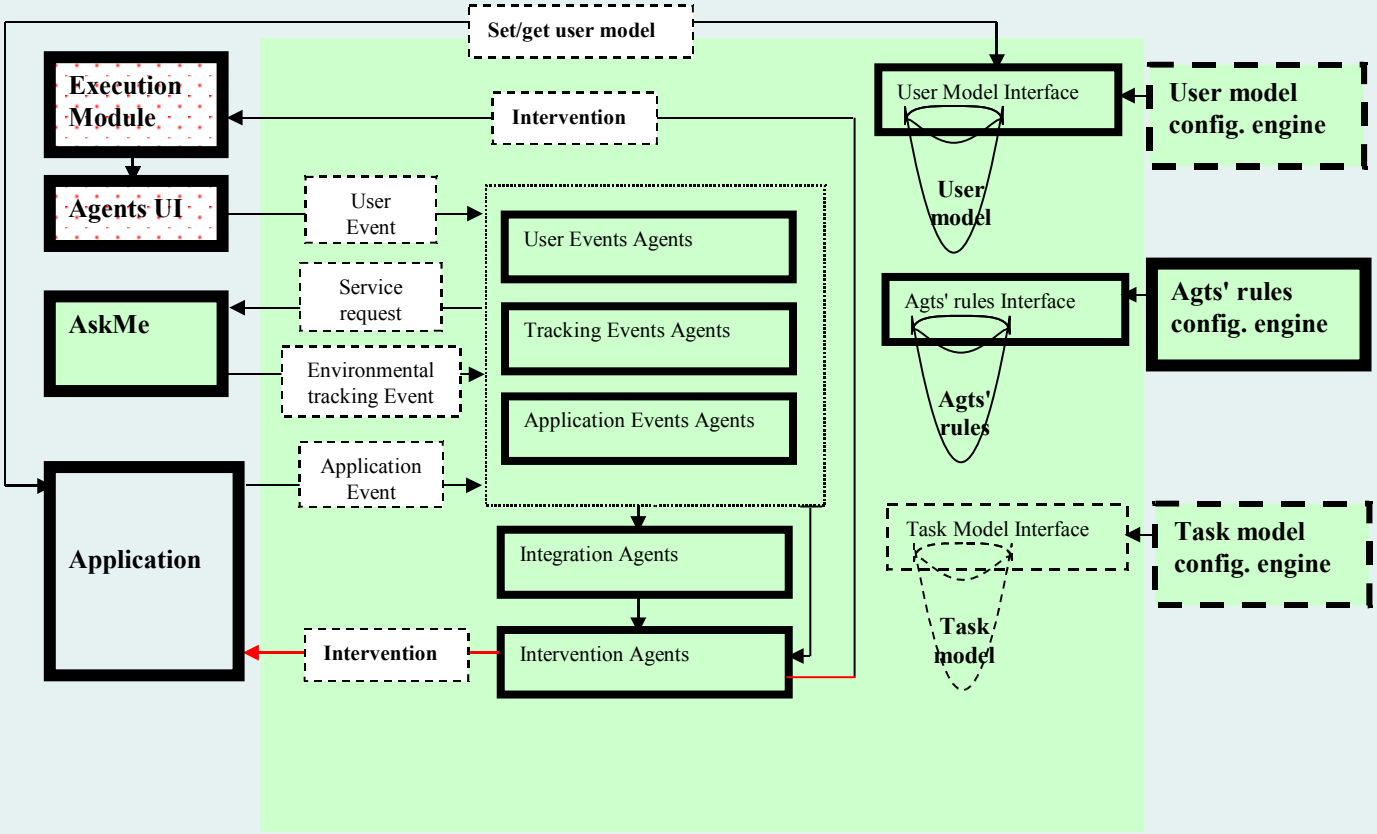
From D3.1



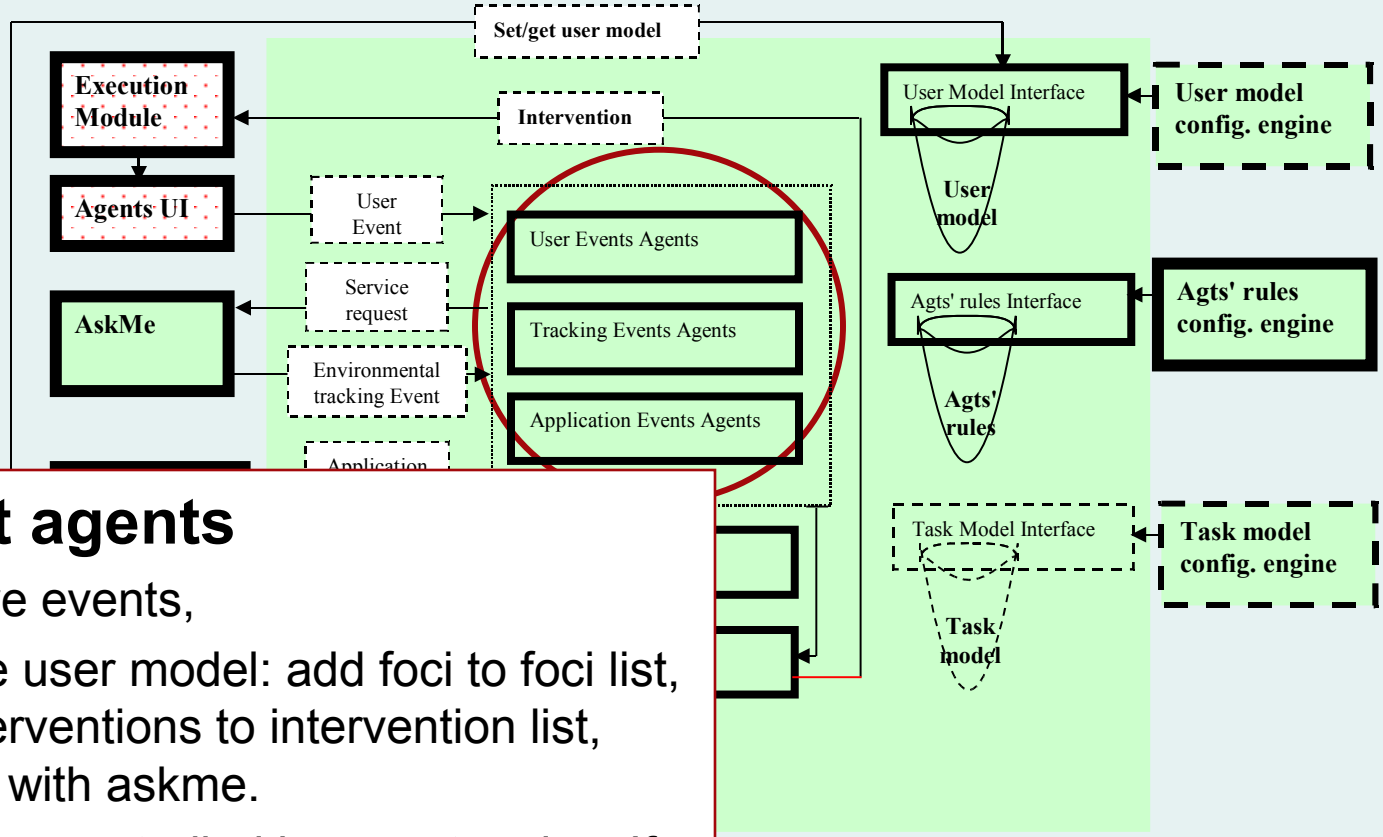
Overview


Set/get user model
Execution Module
Intervention
User Model Interface
User model
Agents UI
User Event
AskMe
Agts' rules Interface
Agts' rules
Agents
Application Event
Application
Intervention

Overview


Set/get user model
Execution Module
Intervention
User Model Interface
User model config. engine
Agents UI
User Event
User Events Agents
User model
AskMe
Service request
Tracking Events Agents
Agts' rules Interface
Agts' rules config. engine
Environmental tracking Event
Application Events Agents
Agts' rules
Application
Application Event
Task Model Interface
Task model config. engine
Integration Agents
Task model
Intervention
Intervention Agents

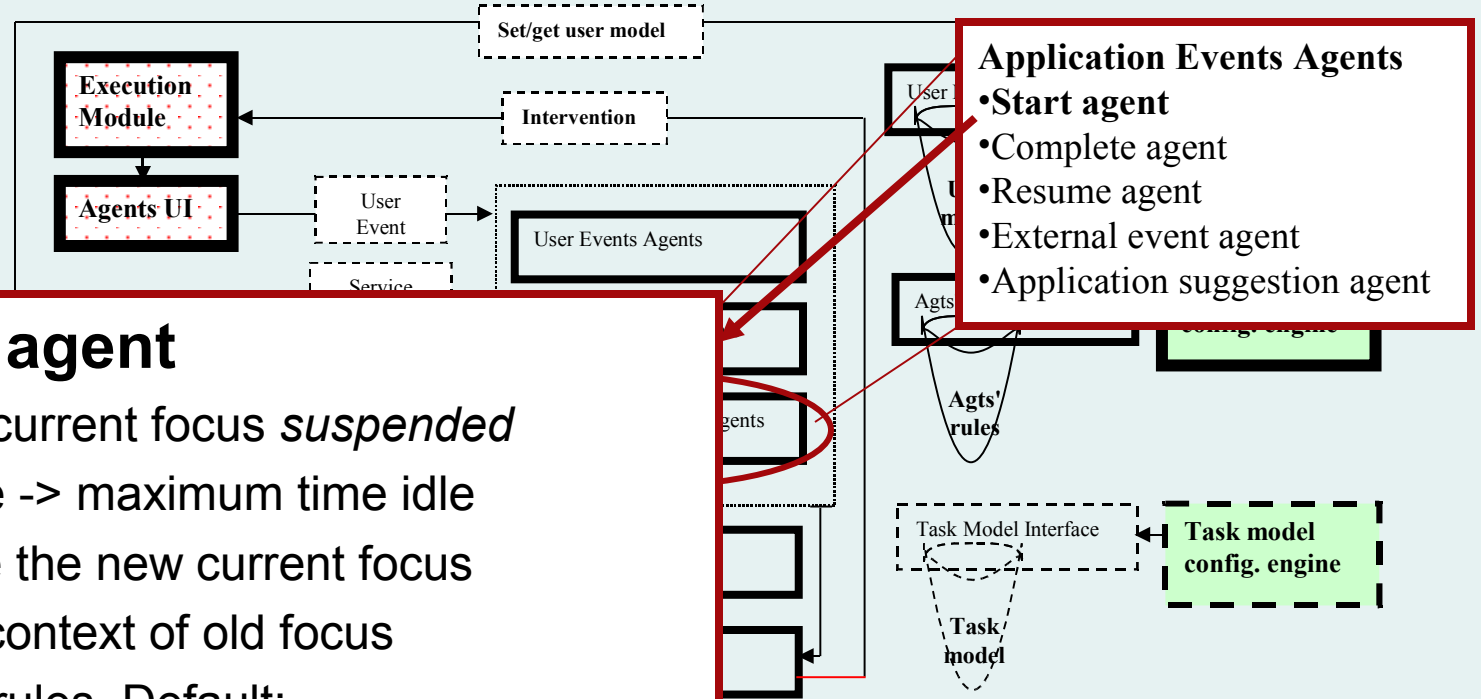
Event agents



Event agents

- Receive events,
- Update user model: add foci to foci list, add interventions to intervention list, interact with askme.
- Behavior controlled by agents rules. If no configuration is done only default rules are used

Event agents (example)



Start agent

- Make current focus *suspended*
- Askme -> maximum time idle
- Create the new current focus
- Save context of old focus
- Apply rules. Default:

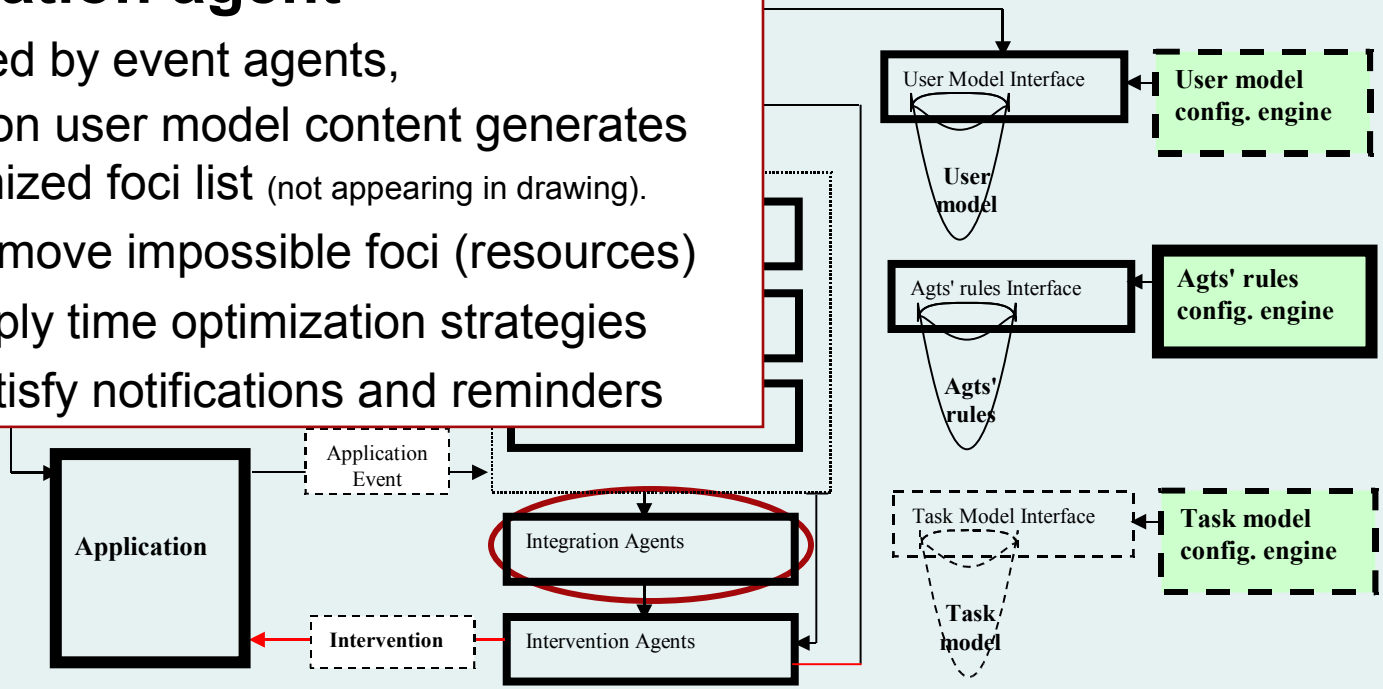
DEFAULT RULE 1: Propose alternative urgent tasks on task start

IF event(start, task(id=T, T_params))
T(urgency)= low
T' / T'(urgency) = high
THEN increase priority of T'
T,T' Task; T' Suspended_Task

Integration agent

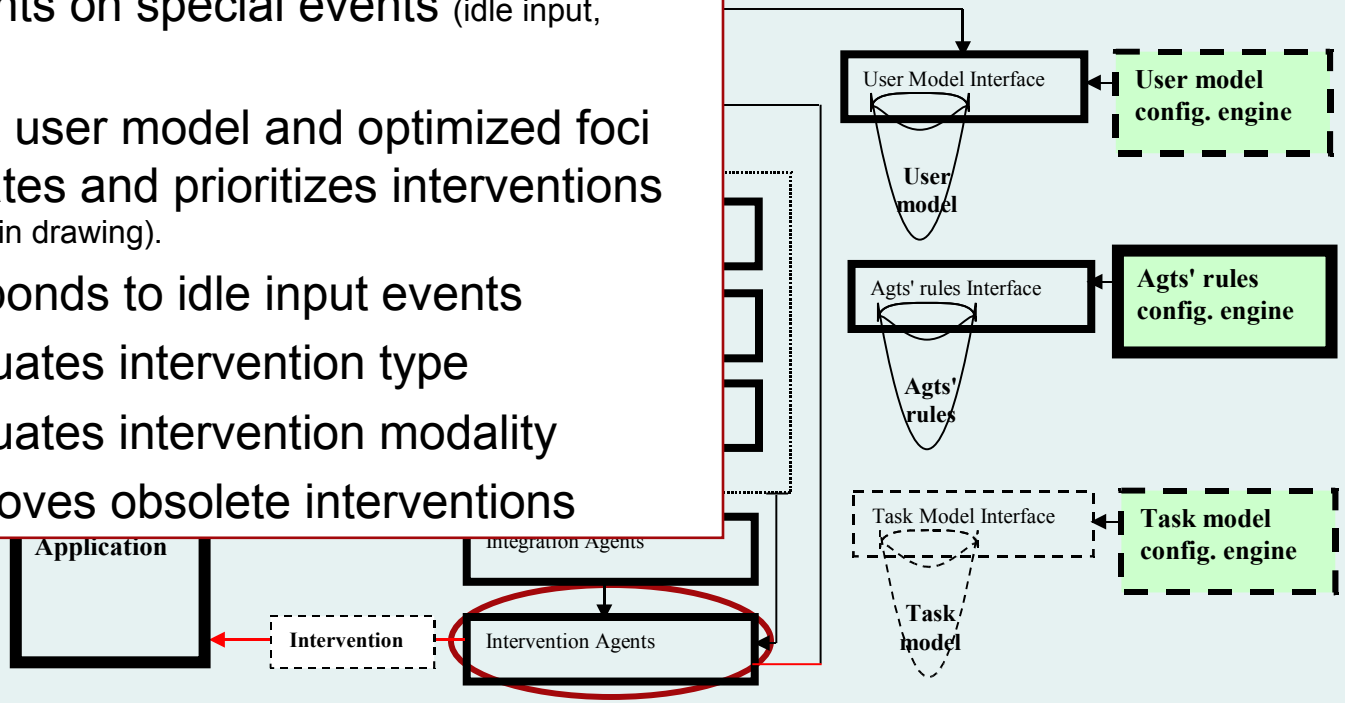
Integration agent

- Activated by event agents,
- Based on user model content generates an optimized foci list (not appearing in drawing).
  - Remove impossible foci (resources)
  - Apply time optimization strategies
  - Satisfy notifications and reminders

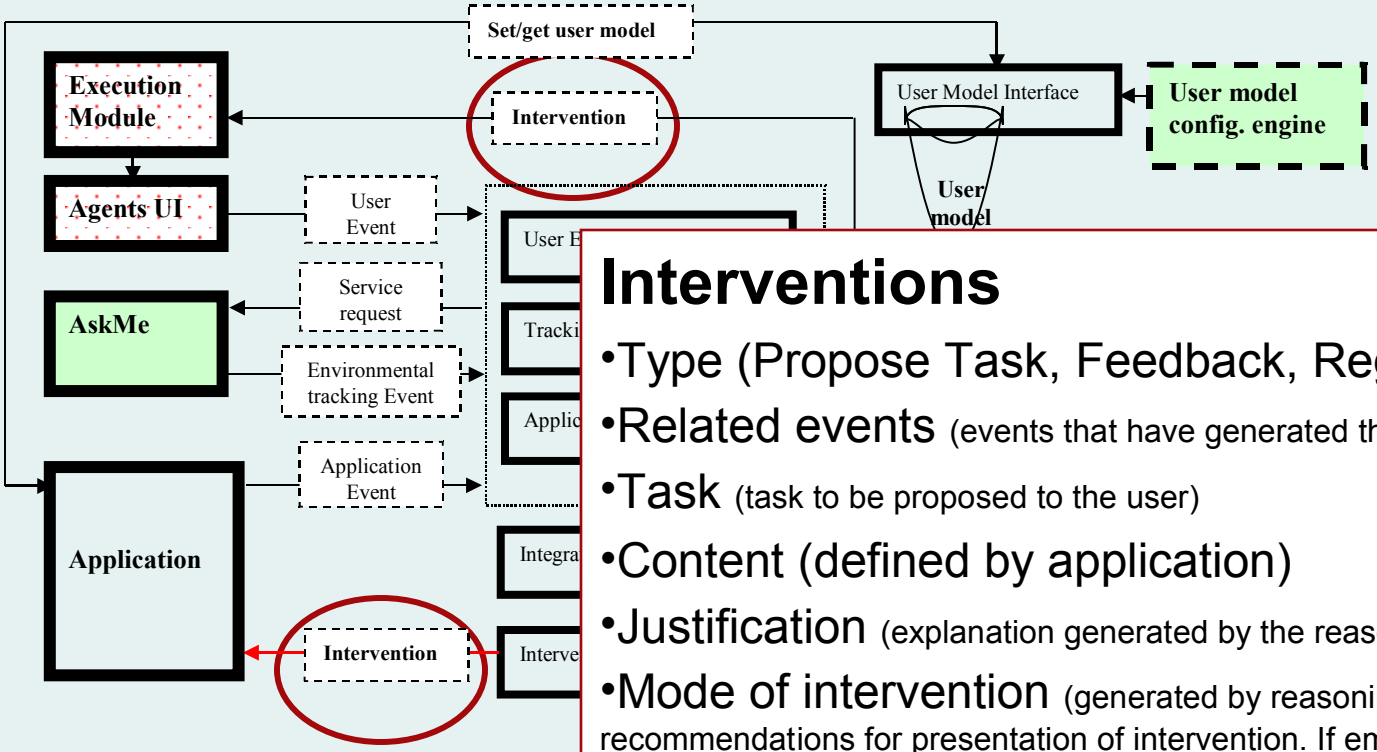


Intervention agent

- Activated by integration agent, or event agents on special events (idle input, breakpoint),
- Based on user model and optimized foci list generates and prioritizes interventions (not appearing in drawing).
  - Responds to idle input events
  - Evaluates intervention type
  - Evaluates intervention modality
  - Removes obsolete interventions



Interventions



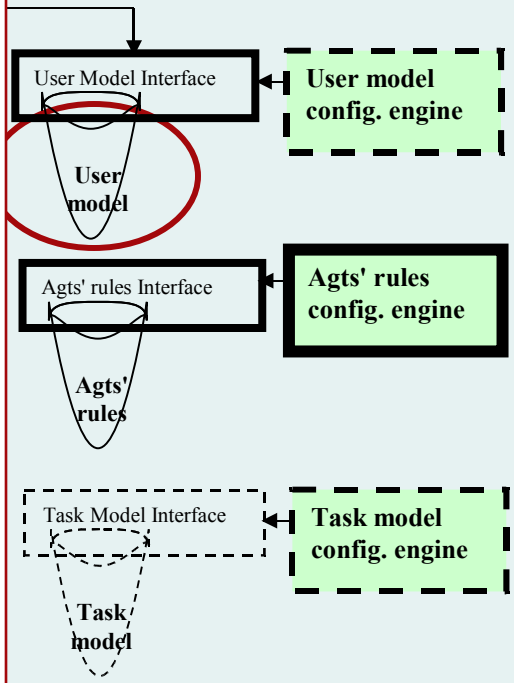
Interventions

- Type (Propose Task, Feedback, Regulation)
- Related events (events that have generated this intervention)
- Task (task to be proposed to the user)
- Content (defined by application)
- Justification (explanation generated by the reasoning module)
- Mode of intervention (generated by reasoning module, recommendations for presentation of intervention. If embodied agent type, this will be the interface to Cantoche module. Indicates media (text, speech, agent, …), strength, etc.

User model

- Foci list (<user-task, priority, state> these are possible alternative foci)
- Intervention list (<intervention, priority, state> these are possible interventions)
- Preferences (maximum frequency of interruption, no-interruption time, notification modalities)
- Notification requests (<event, intervention>)
- Intervention history (<intervention, time, context, user-feedback)
- User-task model (<task, urgency, deadline, continuation, difficulty level, keywords, relevant people, interruption preferences, priority (?), progression, expected duration> based on the generic task description)
- Time available
- Social network
- Reminders requests
- Events history
- …
- DO WE WANT TO ADD LEARNING SPECIFIC ITEMS AS REQUESTED BY REVIEWERS?

del



Task model

Task model (user independent)

- ID
- Type
- Name
- Maximum idle time
- Continuation (following task(s))
- Difficulty level
- Keywords
- Relevant people / social network
- Required resources (new information available events, user state (e.g. logged in), task state, …)
- Explanation to user
- Expected duration