GAS Ontology

An ontology for collaboration among ubiquitous computing devices

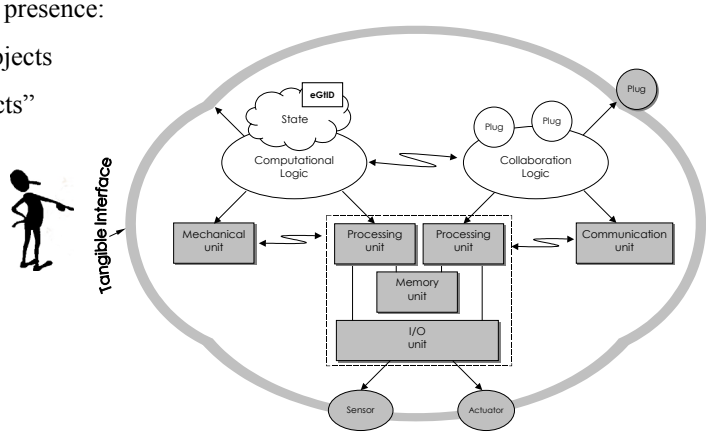
Eleni Christopoulou, Achilles Kameas
Research Academic Computer Technology Institute
Research Unit 3

eGadget

eGadgets (eGts) are everyday tangible objects enhanced with sensing, acting, processing and communication abilities

eGts exhibit a dual presence:

- "tangible" objects
- "digital objects"

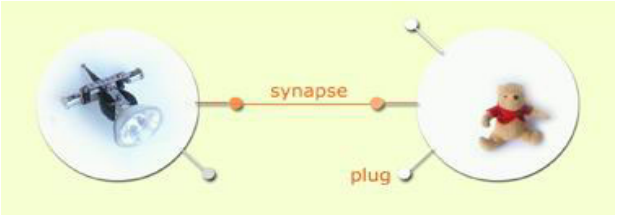


Plug - Synapse

eGts services are manifested as ***Plugs***

Plugs are software classes that make visible eGts capabilities to people and to other eGts

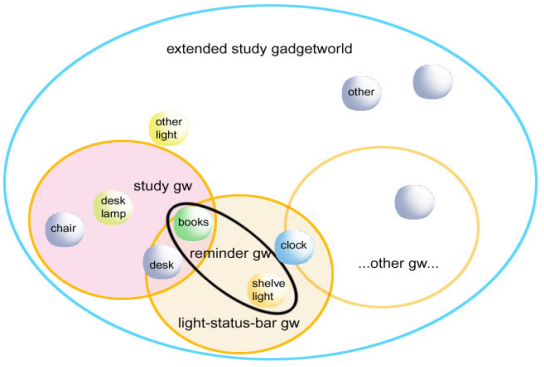
Synapses are associations between two compatible plugs



eGadgetWorld

eGadgetWorlds (eGWs) are dynamic distinguishable, functional configurations of associated eGts, which communicate and / or collaborate in order to realize a collective function

eGWs are formed purposefully by an actor (user or other) and appear as functionally unified entities

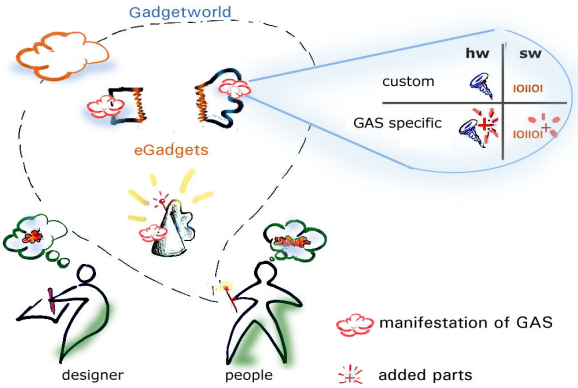


Gadgetware Architectural Style

Gadgetware Architectural Style (GAS) provides the conceptual and technological framework for creating eGWs and using them in a consistent and intuitive way

GAS defines:

- an architectural vocabulary (eGt, Plug, Synapse, eGW)
- configuration rules (for Synapse establishment, eGW storage,…)
- a technological infrastructure (the GAS operating system, ...)



GAS Ontology

GAS Ontology provides the common language for the communication and collaboration among eGts

- Describes the semantics of the basic terms (eGt, Plug, Synapse, eGW)
- Defines the relations among them

GAS Ontology ensures

- eGts replacement feasibility
- Plugs compatibility
- Services discovery

GAS Ontology Infrastructure

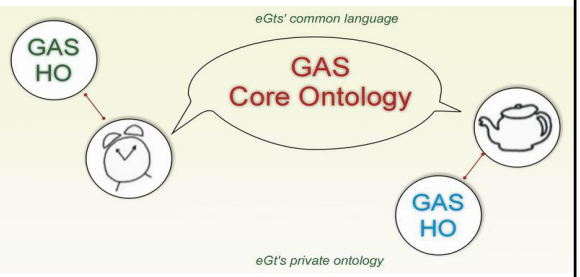
GAS ontology consists of two layers: GAS Core and GAS Higher ontology

GAS Core ontology (GAS-CO)

- describes the semantics of the basic terms
- defines their relations and their roles
- provides the common language among eGts

GAS Higher ontology (GAS-HO)

- contains eGt acquired knowledge
- is eGt private ontology



Building GAS Ontology

DAML+OIL is a semantic markup language

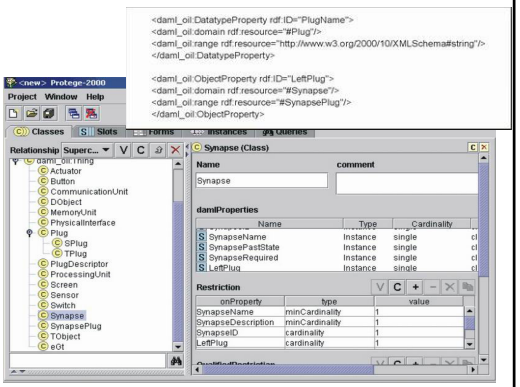
Supports multiple inheritance, property constraints (domain, range, cardinality), default property values

Protégé-2000 is an easy to use graphical interactive ontology editor

Characterized by its scalability and extensibility

Is component-based

Supports storing and importing ontologies in DAML+OIL



Ontology Manager

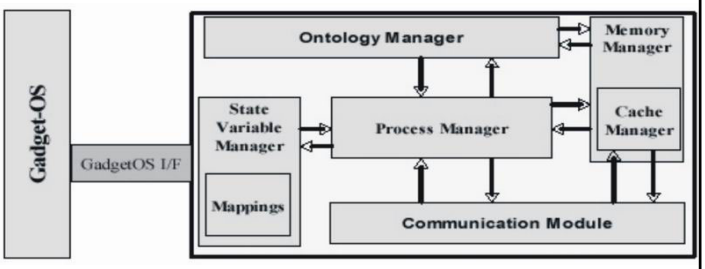
GAS - operating system manages the resources of eGts and enables participation in eGWs through plug and synapse management services

Ontology Manager is responsible for the interaction of the eGt with its stored ontology and the management of this ontology

Edits GAS-HO, making feasible the "storage" and "abstraction" of knowledge

Enables the exchange of knowledge from eGts' ontologies

Composes eGts' queries



eGadgets
www.extrovert-gadgets.net

Fin.

Contact:

Eleni Christopoulou

Computer Technology Institute, Patras, Greece

email: hristope@cti.gr