



AGI engineering

Subscribe



SEMANTIC STORAGE III

INTEGRITY AND CONSISTENCY



Mykola Rabchevskiy

Jul 28 1

An essential aspect of the knowledge storage system (see [Architecture](#)) is maintaining the integrity and, to the extent possible, data consistency. Providing "complete" logical consistency seems unrealistic because of the complexity of the problem and the complexity of determining what is considered such. Nevertheless, it makes sense to prevent the emergence of apparent logical inconsistencies.

Semantic Storage is used as a universal classifier of objects, supplying each logical entity with a set of *signs*. Since the properties of entities can change, sets of the signs (*syndromes*) are modified. As practice shows, one of the potentially frequent contradictions is the *presence of mutually exclusive signs in the syndrome* because the addition of a new sign is not accompanied by the exclusion of an alternative one (both when making changes manually and algorithmically). So *Semantic Storage* has a built-in mechanism to support the manipulation of mutually exclusive signs.

A group of mutually exclusive signs is combined into a *concept* provided with the *MUTEX (MUTually EXclusive)* sign; if one of these features is added to the syndrome of any entity, the alternative feature, if any, is automatically removed.

An example of using mutually exclusive (alternative) signs "C" "M" "I" "VI"

© 2021 Mykola Rabchevskiy. See [privacy](#), [terms](#) and [information collection notice](#)



Publish on Substack

AGI engineering is on [Substack](#) – the place for independent writing

```

AGI: size M MUTEX
AGI: Create 1 entities size and execute? Press y or
AGI:
AGI: S size
AGI: Create 1 entities S and execute? Press y or
AGI:
AGI: M size
AGI: Create 1 entities M and execute? Press y or
AGI:
AGI: L size
AGI: Create 1 entities L and execute? Press y or
AGI:
AGI: XL size
AGI: Create 1 entities XL and execute? Press y or
AGI:
AGI: size
AGI: Execute? Press y or
AGI: size M MUTEX
AGI: size
AGI: Execute? Press y or
AGI: L
AGI: M
AGI: XL
AGI: S
AGI: my_size M
AGI: Create 1 entities my_size and execute? Press y or
AGI:
AGI: my_size
AGI: Execute? Press y or
AGI: my_size M
AGI: my_size L
AGI: Execute? Press y or
AGI:
AGI: my_size
AGI: Execute? Press y or
AGI: my_size L

```

As we can see, with the addition of the "L" sign in the presence of the mutually exclusive sign "M," the last one is automatically deleted.

As far as data integrity is concerned, the most common problem is an erroneous deletion of an entity and erroneous modification of entity syndrome. Two congenital concepts, *IMMUTABLE* and *IMMORTAL*, are provided to prevent such errors. An entity equipped with the *IMMUTABLE* sign does not allow syndrome (set of signs) modification but can be deleted. An entity bearing the *IMMORTAL* sign cannot be removed, but its syndrome can be modified. Entities with both of these signs cannot be removed, and their syndrome cannot be changed. All congenital concepts are protected from both deletion and modification.

Removal of a concept, that is, *an entity used as a sign* (is an element of a syndrome of at least one entity), obviously requires its removal from all syndromes in which it is present. The system performs such removal if none of the entities with the syndrome to be

modified is protected by the IMMUTABLE sign. However, if the concept to be deleted is present in the syndrome of at least one entity protected from modification, the deletion of the concept is blocked. Thus, a *concept present in the syndrome of an entity that cannot be modified cannot be deleted even if it does not have the IMMUTABLE sign*.

[Subscribe](#)[← Previous](#)[Next →](#)

Ready for more?

[Subscribe](#)