LLM CMS Specification Work Offer

This Document is an offer by:

Ing. Wolfgang Scherer, IKT-Beratung,
Wiener Straße 60/11/9,
A-3002 Purkersdorf
Austria, European Union

To

CEIT RALTEC gemeinnützige GmbH,
Concorde Technology Center Schwechat (concorde.tcs)
Am Concorde-Park 2, Gebäude F
A-2320 Schwechat
Austria, European Union

concerning the production of a specification for the central management system (CMS) of the system implementing the service functionality in the Long Lasting Memory (LLM) research project.

About this offer

This offer describes work performed in producing the functional and top level implementation specification for the central management system in the Long Lasting Memories research project. Most important, it describes:

- what will be contained in the specification
- in what level of quality and dependability and detail
- how the above information will be obtained and refined

To this effect, this is a meta-specification, specifying a specification and the associated work.

Document ID	PUCONLLMCMSSOFR001011
Version	V 1.1-0
Author	Wolfgang Scherer
Customer	CEIT Raltec
Customer Approval	Walter Hlauschek
Document State	Proposal
Change Date	2009-12-11
File Name:	LLM CMS Specification Work Offer - 20091211.doc

Overall Target

The target of the specification work is functional and structural specifications that may be given to implementing software development teams to be able to nearly immediate commence implementation. “Nearly immediate” means that implementation must not start before 2 activities deemed necessary at the start of the implementation:

Review of the specification by the implementation team

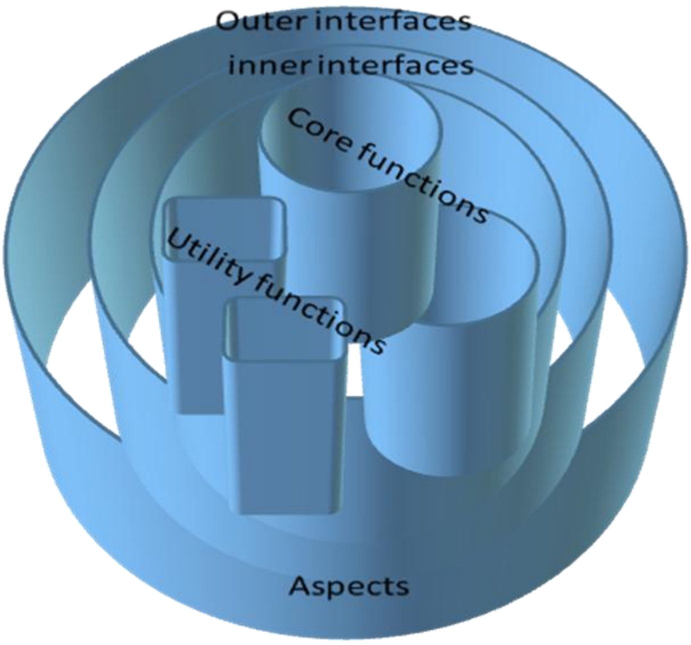
Acceptance and agreement of specification interpretation

Modularity

The specification is intended to be modular in the sense of breaking down the specified systems’ architecture into modules being able to be implemented independently from each other. Furthermore, dependencies of modules amongst each other – however, on a very high level – will be identified in order to allow either an implementation sequence or postponing the implementation of certain modules to a later point of time without compromising the functionality

Template Architecture

In order to estimate the amount of specification work reasonably accurate, the work needs to be broken down into functional entities whose functionality and behavior needs to be specified. However, the architecture of the system is not known at the time of functional specification, much less before specification. In order to break this deadlock situation, this offer assumes an abstract template architecture to identify the class and cardinality of functional entities.



Architectural Entities

Outer Interfaces

Outer Interfaces are interfaces visible from the outside of the system in discussion

Inner Interfaces

Inner Interfaces are interfaces between inner components of the system in discussion. Not all interfaces between components will be considered in the first version of the architecture. However, these of relevance for the respective stage of architecture refinement are listed.

Core Functions

Core functions are the central functionalities of the system in discussion, the reasons of existence for which the system is intended in the first place. Use cases local to a specific core function are also described here.

Utility functions

These are the functions identifiable to core functions and other utility functions, common algorithms and data structures being reusable.

Aspects and Use Cases

These are aspects of the system, functional or non-functional, that are orthogonal to the architectural structure of the system. Aspects may describe performance or quality requirements as well as volumetric. Use Cases are also handled in this category, if they are system-wide.

Cardinalities and efforts

The following table summarizes the assumptions made about the number of architectural elements to be specified in LLM CMS broken down by type of element resulting in a first estimation of effort to generate the specification .

Architectural Element(s)	Number of instances of this class of element
1 - Outer Interfaces **- 1.1- ILC – Independent Living** **- 1.2 - CTC – Cognitive Training** **- 1.3 - PTC – Physical Training** **- 1.4 - Local Interface** **- 1.5 - Remote Interface**	**5**
2 - Inner Interfaces	**6**
3 - Core functions: **- 3.1 - Domain Administration** **- 3.2 - User/Patient Administration** **- 3.3 - ILC control** **- 3.4 - PTC control** **- 3.5 - CTC control** **- 3.6 - Remote/Local Interface control**	**6**
4 - Utility functions	**5**
5 - Aspects and Use Cases: **- 5.1 - Performance** **- 5.2 - Availability** **- 5.3 - Safety** **- 5.4 - Security** **- 5.5 - Modularity** **- 5.6 - Multi-Tenancy / -Agency / -Customer** **- 5.7 - UC: e-Home** **- 5.8 - UC: Day-Care Center** **- 5.9 - UC: Clinical Environment** **- 5.10 - Scalability** **- 5.11 - UC: Home-Care Professionals** **- 5.12 - Accounting** **- 5.13 - Evidence Collection & Archiving**	**13**

Work items

Having sketched the template architecture, each architectural entity requires a certain set of work items to reach the required level of specification.

Item-IDs	Element Type	Part(s) of work	Effort for part (Ph)	Effort for element (Ph)	Number of elements (instances) (Ph)	Total Effort for all instances (Ph)
1.x.y	Outer Interface	1.x.1 - Outer side 1.x.2 - Inner side 1.x.3 - performance/volumetric 1.x.4 - technology aspects	2 2 1 1	6	5	30
2.x.y	Inner Interface	2.x.1 - Outer side 2.x.2 - Inner side 2.x.3 - performance/volumetric 2.x.4 - dependencies 2.x.5 - technology aspects 2.x.6 - modularity aspects	1 1 1 1 1 1	6	6	36
3.x.y	Core function	3.x.1 - Function 3.x.2 - HMI aspects 3.x.3 - Use cases 3.x.4 - configuration 3.x.5 - performance/volumetric 3.x.6 - storage/persistence 3.x.7 - technology aspects	1 1 2 1 1 1 1	8	6	48
4.x.y	Utility function	4.x.1 - Function 4.x.2 - HMI aspects 4.x.3 - configuration 4.x.4 - performance/volumetric 4.x.5 - storage/persistence 4.x.6 - technology aspects	1 1 1 1 1 1	6	5	30
5.x.y	aspect	5.x.1 - General requirements	1	1	13	13
6	Document control and release					8
	Total					**165**

Synergistic Effects, Optionality

In order to minimize effort and cost and, additionally, to structure the whole CMS project modularily, the following options are available:

- Interfaces being completely identical may be specified only once. However, this may not apply to an outer and an inner interface, even if they look the same. Although, specification work and reuse may be employed by sub-structuring.

- The CMS consists of mandatory components and optional ones. Optional components are identified below and also marked with dependencies (an optional component may become mandatory if another optional component is requested but requires the former one)

- Outer Interfaces: ILC,PTC and CTC interface may be specified as being identical, their different properties differentiated only in the core functions section of ILC/CTC/PTC-control

- Core functions: domain admin may be optional, however, it will most probably contain general functions best let be general and therefore may save complexity in other core functions. This decision, though, can most probably be made at the high level design stage and, unfortunately not in the specification phase.

- Core functions: ILC/PTC/CLC control could be made the same, saving 2 of 3 specification steps. However, the gain in specification simplicity may result in usability issues and the different object models for ILC/PTC/CTC must be denoted somewhere if not supported be the reflexive properties of the ILC/CTC/PTC interfaces, which is not to be expected.

- Aspects: Multi-Tenancy may be optional, Scalability and Accounting may be made optional and treated in a later iteration of the specification and system implementation.

- Administrative work, however, increases in effort when the specification work is divided into multiple parts. This has its reason in the fact, that each work part will deliver a new, separate version of the specification. The work item 6 – “document control” will have to be repeated for each issue/modular part.

Orderable Specification Parts

From the above, the following specification parts can be ordered:

1. Base Minimum Specification:

outer interfaces:	1.1=1.2=1.3; 1.4=1.5 => 2 instances	=> 12 Ph
inner interfaces:	only 3 instances assumed	=> 18 Ph
core functions:	all above	=> 48 Ph
utility functions:	only 3 instances assumed	=> 18 Ph
aspects:	all above	=> 13 Ph
Base document control:		=> 8 Ph
Total base specification:		== 117 Ph => € 7605.-

2. One additional outer interface: => 6 Ph => € 390.-

3. One additional inner interface: => 6 Ph => € 390.-

4. One additional utility function: => 6 Ph => € 390.-

5. Document control per specification issue: => 8 Ph => € 520.-

As for dependencies, the above split model does not impose any dependencies as of the time of writing.

References

- “Auszug_LLM_DoW_v3 1_approved_10042009.pdf” of 2009-12-02