A draft of the IM Engine SPI feature description list

Disclaimers

Each feature may be implemented by one or more interfaces, and each interface may implement one or more features.

Feature: Initialize/Finalize

No.	Function	Can be called by IM Engine	Called by User Interface component	Hooked by IM framework	Required/Given Information	Return/Retrieved Information	Description / Behavior
1	Loading IM Engine Module			O	IM engine shall provide name of engine, locales supported by this IM Engine, User Interface Components used by the IM engine and IM logics supported. A method table shall be provided by both ends.	Success or fail to load	Engine can be loaded when session does not even exist. Therefore engine instance and session association will not be done at the engine loading time. In cascade case, IM engine is virtual engine as acting proxy. It should provide a method to retrieve information from cascaded IM engine. There may be some timing gap in initializing actual engine.
2	Unloading IM Engine Module			O	IM engine identifier	Success or fail to unload	IM engine container may not attempt to unload IM engine without finalizing IM engine.
3	Initialize IM Engine			O	Session Identifier	Success or fail to initialize	Initialization can be called many times as multiple session initialize, and maybe call some of session reset and some of the component restart. IM engine is able to do something for initialization and Initialization should not bound to engine loading. Session identifier should be definitive argument.

4	Finalize IM Engine			O	Session Identifier	Success or fail to finalize	Finalization should not be bounded to engine unloading. IM Engine is able to do something for finalization of the session. A session identifier should be a definitive argument. If IM engine allocates some memories use convenient function to manage memory, it should free these memories.
5	Create Input Context			O	Input Context Session Identifier	Success or fail	Input Context is created by IM Framework prior to calling this function. IM Engine should create corresponding internal data structure, storing some data needed for inputting such as IM logic's ID etc. and finish initialization work by using IC identifier and Session Identifier. For multiple functions inside IM engine to communicate over IC, IC should have some mechanism to pass multiple sets of data, such as tags, session identifier, etc.
6	Destroy Input Context			O	Input Context Session Identifier	Success or fail	Notify that Input Context is about to be deleted by IM Framework. Input Method should free the private data corresponding to this Input Context and do some cleanup work.

Feature: framework/engine negotiation

7	Setting Information	O	O	O	Specify object identifier to set information; IM engine identifier , IC Identifier, session identifier, and IM logic identifier and others. Information to be set.	Success or fail	This is negotiation framework. It should not bound to static sequence. It should be independent and bi-directionally inform/retrieve necessary information when it need. It can work on per session.
8	Getting Information	O	O	O	Specify object identifier to retrieve information; IM engine identifier , IC Identifier, session identifier, and IM logic identifier and others.	Information to be retrieved.	This is negotiation framework. It should not bound to static sequence. It should be independent and bi-directionally inform/retrieve necessary information when it need. It can work on per session.

Feature: mtext Utility Functions

No.	Function	Can be called by IM Engine	Called by User Interface component	Hooked by IM framework	Required/Given Information	Return/Retrieved Information	Description / Behavior
9	mtext handling functionality	O	O		mtext(m17nlib) to be inputted.		Adopt text handling framework such as from m17nlib. Conversions from/to keyevent should be supported. This includes functionality to set additional information to mtext for assisting or hinting user's inputing such as clause, pronunciation, annotation, etc. Application and IM engine should be able to negotiate supported additional information.

Feature: Event Flow							
No.	**Function**	**Can be called by IM Engine**	**Called by User Interface component**	**Hooked by IM framework**	**Required/Given Information**	**Return/Retrieved Information**	**Description / Behavior**
10	Setting Focus			O	Input Context	None	Notify that an Input Context gets focus. This function should address issues caused by incoming/outgoing queue delays when focus changes state.
11	Unsetting Focus			O	Input Context	None	Notify that an Input Context loses focus.
12	Receiving synchronous event from the IM Client			O	Session identifier Input Context Event	Success or fail	Client sends a KeyPress or KeyRelease event, or other synchronous event to corresponding Input Context.
13	Send synchronous event to the IM Client	O	O		Session identifier Input Context Event	Success or fail	

14	Switching IM Logic	O	O			None	IM Logic notifies to an IM Framework that the IM Logic wants to change to another one. This functionality will be necessary for IM Module which supports multiple IM Logics. Switching need to be beyond container and module. UI to switch the logic able to be quite easy to go beyond container boundary, because of UI component knows all information. Engine container initiated to logic switching beyond engine container boundary requires, other engine container or UI component. How to realize it to be determined. ex. Generic shortcut for UI component. AI to investigate possible method to switch from one to other logic inside or beyond engine container. As Quickly conversion on/off is occurred. ex. European people. Client side library should handle logic switch transparently inside IMBUS and need to handle profile.
15	Starting Pre-edit Operation	O			Input Context	None	Notify to an IM Framework that IM Engine needs to start Pre-edit operation.
16	Stopping Pre-edit Operation	O			Input Context	None	Notify to an IM Framework that IM Engine doesn't need Pre-edit operation
17	Drawing Pre-edit	O			Input Context mtext to be drawn	None	Notify to an IM Framework that IM Engine needs to draw pre-edit text. This function needs Input Context and mtext to be drawn as a pre-edit string/attribute.
18	Move Caret Position of Pre-edit string	O		O	Input Context Caret Position	None	Notify Caret Position of Pre-edit string is changed. This function can be called by both IM Framework and IM Engine.

19	Starting Status Operation	O			Input Context	None	Notify to an IM Framework that IM Engine needs to start Status drawing operation. IM engine should be able to handle multiple IM status instances associated with different UI component simultaneously, including traditional status and panels.
20	Stopping Status Operation	O			Input Context	None	Notify to an IM Framework that IM Engine doesn't need Status drawing operation.
21	Drawing Status	O			Input Context Text to be drawn.	None	Notify to an IM Framework that IM Engine needs to draw Status string.
22	Event Control	O			Input Context	None	Notify to IM engine container that IM engine wants to take control of event flow, for example to implement semantic IM on/off.

Feature: Lookup Choice							
No.	**Function**	**Can be called by IM Engine**	**Called by User Interface component**	**Hooked by IM framework**	**Required/Given Information**	**Return/Retrieved Information**	**Description / Behavior**
23	Setting/getting Information about Lookup Choice	O			Input Context Pointer which holds a layout of Lookup Choice Window	Success or fail	IM Logic notifies information how to display Lookup Choice Window to an IM Framework. The information contains such as a layout of lookup choice window, a title of lookup choice window, page size, selected candidate, current page, etc.
24	Starting Lookup Choice Operation	O			Input Context	None	Notify to an IM Framework that IM Engine needs to start Lookup Choice Operation
25	Drawing Lookup Choice	O			Input Context Candidate List	None	
26	Setting Focus to one of the Lookup Candidates			O	Input Context Index of the Candidates to be focused	None	Change the selected candidate. When framework notifies engine, engine may change selected candidate.
27	Move Lookup Page to Previous			O	Input Context	None	
28	Move Lookup Page to Next			O	Input Context	None	

29	Update Lookup Page Size			O	Input Context Page Size	None	Notify that the client changed an number of candidates displayed for one page. It's an event which sent by IM framework to IM Logic to inform IM Logic the maximum page size that UI can display. IM framework won't change the page size of the candidate list by itself. It's IM Logic's responsibility to change the page size according to the event and update candidate list by feature 41.
30	Stopping Lookup Choice Operation	O			Input Context	None	Notify to an IM Framework that IM Engine has finished Lookup Choice Operation.

Feature: UI Component							
No.	**Function**	**Can be called by IM Engine**	**Called by User Interface component**	**Hooked by IM framework**	**Required/Given Information**	**Return/Retrieved Information**	**Description / Behavior**
31	Start User Interface Component	O			Input Context User Interface's ID async flag	Success or fail	Starting User Interface Component. This function needs an input context and user interface component's id. When UI component finishes initialization, the UI component will notify IM engine that it is ready using feature 12. Function can be either synchronous or asynchronous.
32	Send data to User Interface Component	O			Input Context User Interface's ID data async flag	Success or fail	Sending data to the User Interface Component specified by User Interface Component's ID

33	User interface components to send data to a specific Input context of an IM engine		O		Input Context IM logic ID Data async flag	Success or fail	Sending data to the IM logic specified by the IM logic ID. IM engine can use this to accommodate other input sources besides keyboard, such as handwriting and voice recognition
34	Stop User Interface Component	O			Input Context User Interface's ID async flag	Success or fail	Stopping User Interface Component. This function needs an input context and User Interface Component's ID.
35	Integrate IM Bus events to UIC event loop		O				For the detail to be determined.
36	Send a keyboard event from UIC to an input context		O		Input Context Keyboard Event	None	NOTE) Feature 36 and 37 are useful to implement applications like on-screen keyboard etc.
37	Send keyboard event to an application client		O		Input Context Keyboard Event	None	NOTE) Feature 36 and 37 are useful to implement applications like on-screen keyboard etc.
38	Commit a string to an application client	O	O		Input Context mtext	None	User Interface Component can commit a string to an application client via an input IC without interfering with IM Engine.
39	Initialize and run the User interface component			O	Session ID	None	An user interface component will run in a separated process.
40	Finalize the User interface component			O	None	None	This function will finalize and exit the running user interface component.
41	Attach an input context to the user interface component			O	Input Context	None	This function inform the user interface component that an input context has requested to use this component.

42	Detach an input context from the user interface component			O	Input Context	None	inform that the input context won't use this component anymore.
43	Register the capability of IM engine, UI Component and IM Framework	O	O	O	Capabilities	None	To register the capabilities of IM Logic, User Interface Component and IM Framework, such as the types of objects it can handle (JAR, XUL, XAML, CLR); and storage manager.

Feature: Storage manager							
No.	**Function**	**Can be called by IM Engine**	**Called by User Interface component**	**Hooked by IM framework**	**Required/Given Information**	**Return/Retrieved Information**	**Description / Behavior**
44	Create Storage Manager Context	O	O		Identifier chosen by IM Logic	Storage Manager Context	Create Storage Manager Context which capsulizes system dependent functionality such as read/write files and set engine information for storage manager.
45	Get information from storage manager	O	O		Storage Manager Context Path information	Information to be retrieved.	Retrieve information from storage manager. The informations to be retrieved may be key-value pair or more complex data structure such as XML.
46	Set information to storage manager	O	O		Storage Manager Context Path information information to be set	Success or fail	Set information to storage manager. The informations to be retrieved may be key-value pair or more complex data structure such as XML.
47	Destroy Storage manager Context	O	O		Storage Manager Context	None	Destroy Storage Manager Context.

Feature: IM client text handling

No.	Function	Can be called by IM Engine	Called by User Interface component	Hooked by IM framework	Required/Given Information	Return/Retrieved Information	Description / Behavior
48	Getting IM Client's Text for handling Text Conversion	O	O		Input Context mtext to be converted	Success or fail	Caller notifies IM Framework that the caller needs IM Client's Text for Text Conversion. IM Client should return the Text with a range specified by the caller.
49	Setting IM Client's Text for handling Text Conversion	O	O		Input Context converted mtext	Success or fail	Caller notifies IM Framework that the IM Client needs to receive text as a result from text Conversion.

Feature: Hotkey

No.	Function	Can be called by IM Engine	Called by User Interface component	Hooked by IM framework	Required/Given Information	Return/Retrieved Information	Description / Behavior
50	Create Hotkey from String	O	O		Key's name	Hotkey	Creating Hotkey Structure from String
51	Create Key event from String	O	O		Key's name	Key	Creating Key Structure from String
52	Create Hotkey Mapping Profile	O	O		Input Context	Hotkey Mapping Profile ID	Creates a new Hotkey Mapping Profile within IM Framework. The Hotkey Mapping Profile ID is allocated by the IM Framework.

53	Register Hotkey to the IM Framework	O	O		Hotkey Mapping Profile ID Hotkey List Action	Boolean value	Register list of Hotkeys with corresponding actions into a Hotkey Mapping Profile to IM Framework. The Mapping Profile ID is defined by the IM engine, and each Mapping Profile can contain one or more Hotkeys for one or more actions.
54	Change Hotkey Mapping Profile	O	O		Input Context Hotkey Mapping Profile ID	Boolean value	Notify to an IM Framework that the specific Input Context needs to change the active Hotkey Keymap Profile.
55	Delete Hotkey Mapping Profile	O	O		Hotkey Mapping Profile ID	Boolean value	Deletes the Hotkey Mapping Profile from the IM Framework.
56	Resetting Input Context			O	Input Context	Flushed string	Notify a client needs to reset the Input Context. When IM Engine needs to commit some string, the text should be returned.
57	Register properties to common user interface component, such as panel.	~~O~~	O		Input Context A list of properties to be registered.	None	IM framework takes charge of displaying the properties.
58	Update the information of a property	O	O		Input Context The information of the updated property	None	
59	Trigger a property			O	The id of the property which was triggered by user.	None	When a user trigger a registered property on the panel, this function will be invoked.
60	Notify feedback event to IM framework	O			Input Context operation	None	Notify that IM Engine wants to alert a client.

61	Getting IM engine information			O	IM engine identifier	IM engine information	IM Framework may retrieve IM engine specific information such as amount of IM logics in an engine or IM logic's information anytime after loading module. IM engine information may contain IM engine's information and set of IM logics' information. NOTE)This function should be a part of service discovery functionaries of IM-BUS.

TOTAL 61 features