S@T 01.60 V1.0.3 (2001-06)

Technical Specification

S@TGG

S@T Gateway Implementation Guidelines

1. History

Document history		
Release	Approved by	Comment
V1.0.0	S@T-TDG WG2 #22	First internal release.
V1.0.1	S@T-TDG WG2 #29	Blocking mechanism for forbidden tag values inserted as agreed during WG2 #28.
V1.0.2	S@T-TDG WG2 #29	Minor editorial changes. First approved release for publication.
V.1.0.3	S@T-TDG WG2 #30	Editorial changes, minor fixes for publication

TABLE OF CONTENTS

1. Scope

The present document defines guidelines for the conversion of the S@T Markup Language (SATML) to the S@T Byte code (SBC) by a decoder/encoder (DE) of a S@T gateway (GW).

Defining the guidelines is to ensure interoperability between gateways and SIM byte code browsers independently of the respective manufacturers and operators.

The present document provides

- a system overview,
- recommendations to ensure interoperability and implementation guidelines for SATML/SBC translation *without* providing a 1:1 mapping of SATML elements and attributes to SBC TL[A]Vs. Different SBC command sequences may result either in the same or a comparable browser behaviour,
- a reference sample.

It is not in the scope of this document to issue mandatory rules for SATML/SBC translation to ensure a common "look and feel" of a given SATML service between browsers of different manufacturers. I.e. this document shall ensure *technical interoperability* (product interworking) between gateways and browsers, however not an identical behaviour as far as the MMI is concerned.

2. Normative References

[GSM 11.14]	GSM 11.14: "Digital cellular telecommunications system (Phase 2+); Specification of the SIM Application Toolkit for the Subscriber Identity Module - Mobile Equipment (SIM - ME) interface"
[GSM 03.38]	GSM 03.38: "Digital cellular telecommunications system (Phase 2+); Security Mechanisms for the SIM application toolkit".
[ISO8879]	"Information Processing - Text and Office Systems - The Generalised Markup Language (SGML)", ISO 8879:1986.
[WML 1.1]	Wireless Application Protocol - Wireless Markup Language (WML) 1.1
[WTAI 1.1]	Wireless Application Protocol - Wireless Telephony Application Interface Specification (WTAI) 1.1
[WTAIGSM 1.1]	Wireless Application Protocol - Wireless Telephony Application Interface Specification - GSM Specific Addendum (WTAI GSM) 1.1
[RFC2396]	"Uniform Resource Identifiers (URI): Generic Syntax", T. Berners-Lee et al., August 1998. URL: ftp://ds.internic.net/rfc/rfc2396.txt
[SATML1.0]	SATML, S@T Markup Language (Technical Specification S@T 01.00)
[SBC1.0]	SBC, S@T Byte Code (Technical Specification S@T 01.00)

3. Definitions and Abbreviations

3.1. Definitions

card A card element specifies one unit of navigation.

deck A deck is the smallest unit that can be sent to a SIM browser using SSP.

may This word or the adjective *optional* means that an item is truly optional. One vendor may choose to include the item because a particular marketplace requires it or because the vendor feels that it enhances the product, while another vendor may omit the same item. An implementation that does not include a particular option *must* be prepared to interoperate with another implementation that does include the option, though perhaps with reduced functionality. In the same vein, an implementation that does include a particular option *must* be prepared to interoperate with another implementation that does not include the option (except, of course, for the feature the option provides).

must This word, or the terms *required* or *shall* means that the definition is an absolute requirement of the specification

must not This phrase, or the phrase *shall not* means that the definition is an absolute prohibition of the specification

service A service is a set of decks that are browsed successively with interactions to the user.

shall see *must.*

shall not see *must not*.

should This word, or the adjective *recommended* means that there may exist valid reasons in particular circumstances to ignore a particular item, but the full implications must be understood and carefully weighed before choosing a different course.

should not This phrase, or the phrase *not recommended* means that there may exist valid reasons in particular circumstances when the particular behaviour is acceptable or even useful, but the full implications should be understood and the case carefully weighed before implementing any behaviour described with this label.

3.2. Abbreviations

DE Decoder / Encoder

DTD Document Type Definition

GW Gateway

HTTP Hypertext Transfer Protocol

M Mandatory

ME Mobile Equipment

MMI Man Machine Interface

O Optional

SAT SIMalliance Toolbox

SBC SAT Byte code

STLS S@T Transport Layer Security, based on GSM 03.48 security

SGML the Standardised Generalised Markup Language (defined in [ISO8879]) is a general-purpose language for domain-specific markup languages

SATML S@T Markup Language

SMSP Short Message Service - Point to Point

SSAL SIM Secured Application Layer

SSL Secure Socket Layer

SSP S@T Session Protocol

TL[A]V Tag Length [Attributes] Value

RFU Reserved for Future Use

STK SIM Application Toolkit

URI Uniform Resource Identifier [RFC2396]

URL Uniform Resource Locator [RFC2396]

WML Wireless Markup Language

XML Extensible Markup Language. XML is a restricted subset of SGML.

4. System Overview

4.1. Architecture

4.1.1. S@T browsing protocol stack

S@T Gateway

SATML/SBC en/decoding	
HTTP	SSP
SSL	STLS
TCP/IP	SMSP

S@T Browser

SSAL
SSP
STLS
SMSP

4.1.2. S@T single card architecture

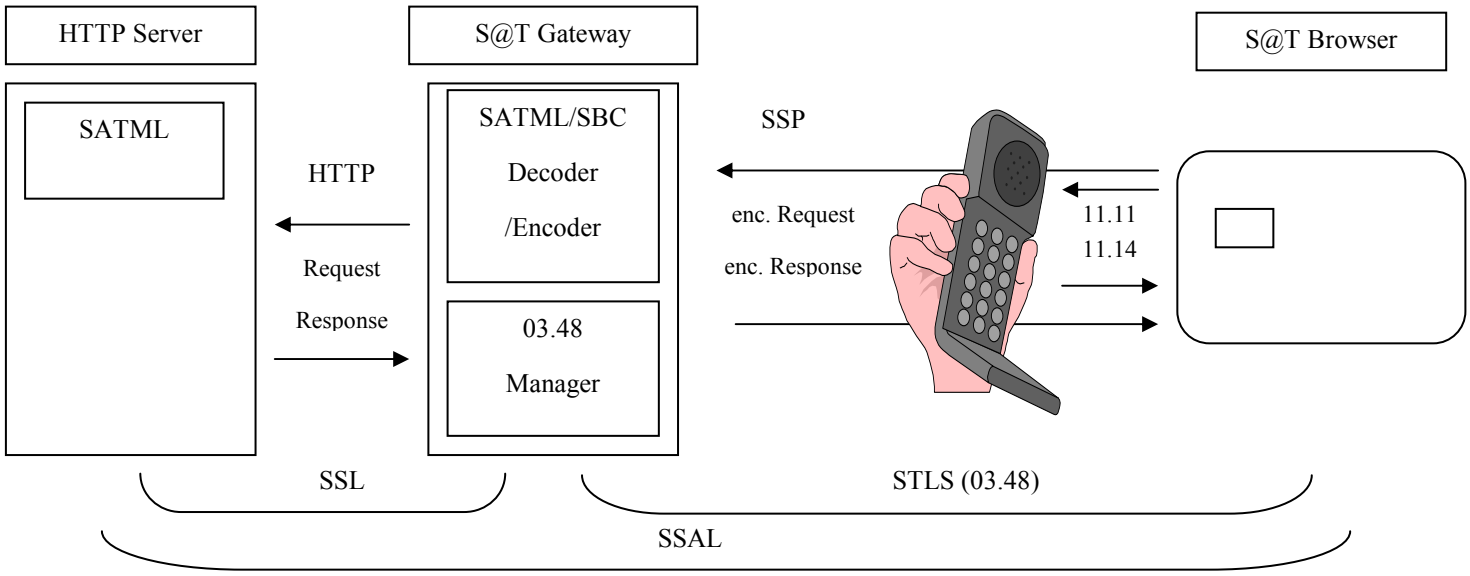

Figure 1: System overview.

4.2. Interoperability Requirements

4.2.1. Session start up sequence

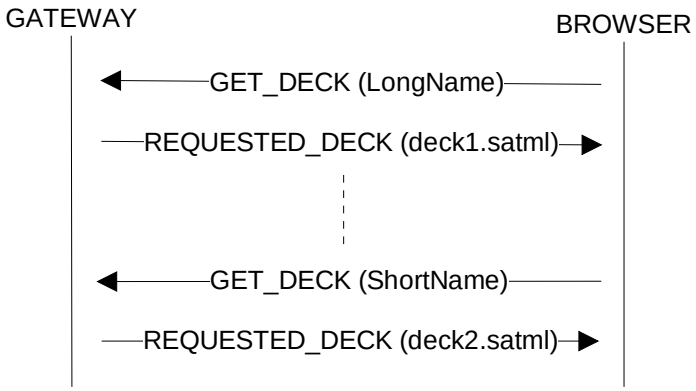

Figure 2: Session start up sequence.

(1) Browser requests deck from gateway. The initial browser request must not contain coded URL references, because coded names may only be valid for a single browsing session. Therefore DE would not be able to decode the short name of the initial request to a full name.

(2) Requested deck will be sent by gateway to browser. All address references to online decks shall be encoded by DE to short name references. The gateway must keep a look-up table to be able to decode browser requests of coded name references to full URL references during a browsing session. However, address references to installed decks (namespace 'sim') must not be encoded to short name references (see 5.6.5 `go`).

(3) Before another deck request is issued to the gateway the browser should look in the cache, if deck is already installed. If not, request with short name reference should be sent to the gateway.

(4) Gateway must decode short name reference to full URL reference, fetch the requested deck either from a gateway´s cache or perform a http request to a content server (see Figure 1: System overview.) and finally send the requested deck to the browser.

Multi-session support is optional.

4.2.2. Bookmarks

S@T 1.22 describes two ways to handle bookmarks. Either the bookmarks are stored in the S@T browser or in a database at the gateway. The commands to set and get bookmark entries are different for both mechanisms. To ensure interoperability the two mechanisms must be implemented for a gateway.

4.3. Data Coding Schemes

4.3.1. Autodetection of Character Encodings

It is assumed that the gateway automatically detects the required target data coding scheme by examination of the input data. ME capabilities (e.g. support of UCS2 encodings) must be taken into account before the target data coding scheme for a respective deck is chosen automatically by a S@T gateway. The autodetection mechanism is gateway implementation dependent, e.g.:

- examination of HTTP response header
- examination of XML header

- examination of characters contained in the SATML source. DCS decision may be based on whether encoding of these characters to SMS default alphabet is possible or not.

It is highly recommended that the gateway takes UCS2 encoding only if really necessary. Otherwise bandwidth would be wasted.

4.4. URLs of Resident Decks

If the DE processes an URL beginning with the prefix 'sim:', DE has to

- remove 'sim:' from the URL
- encode the URL using SMS default alphabet. It is forbidden to use a CodedDeckName in the address reference
- set the Forced Resident attribute of the URL Reference TLAV structure to 1.

4.5. White Space Handling

SATML white space and line break handling is based on [XML] and assumes the default XML white space handling rules for text. DE shall ignore all insignificant white space in elements and attribute values, as defined by the XML specification. White space immediately before and after an element is ignored. In addition all other sequences of white space must be compressed into a single inter-word space.

DE must remove any white spaces appearing in attribute values of the %hex-t; type before conversion to SBC bytearrays, e.g.:

SATML Source

```
<satml>
  <card>
    <sat-gen-stk
    sat-cmdtype=”21”
    sat-cmdqual=”80”
    sat-destdev=”02”
    sat-data=”8D0C0448656C6C6F20576F726C64”/> <!--Hello World-->
  </card>
</satml>
```

SBC Command

														H	e	l	l	o		W	o	r	l	d
D0	17	01	03	01	21	80	02	02	81	02	8D	0C	04	48	65	6C	6C	6F	20	57	6F	72	6C	64

4.6. Return of WTAI Functions

To be able to return to the calling card after a WTAI function has been executed, a card id may be generated by the DE if not already present. Then it is possible to issue a reference to the calling card within SBC.

4.7. Variables

4.7.1. Initialisation of Variables

If an undefined variable is referenced it results in the substitution of an empty string.

4.7.2. Scope of Variables

Temporary Variables

Session valid.

Permanent Variables

Infinite valid until changed by an administrative command. Administrative commands can only be issued by the network operator.

4.7.3. Constraints

Example:

It is impossible to mediate this kind of WML source code:

```
<satml>
     <card id="1">
          <p> Hello World </p>
          <do type = "accept">
              <go href = "#2">
               <setvar name="toto" value="http://www.abc.com?X="/>
               <setvar name="titi" value=""3?y"/>
              </go>
          </do>
     </card>
     <card id="2">
          <p> Bad sample </p>
          <do type = "accept">
              <go href = "$toto$titi = 4"/>
          </do>
     </card>
</satml>
```

because in SBC the address part for the Address Reference TLV and the parameter part for the Parameter Reference TLV structure would have to be extracted from the value of the href attribute at runtime. That is impossible.

4.7.4. Variable Declaration Error Handling

SBC allows allocation of 128 temporary variables (see [SBC1.0]). For the case that DE has declared within a session all variables yet, DE shall send the encoded deck with an error message to the SAT Browser if additional variables need to be declared.

4.8. Security

Access to SATML STK Extension elements shall be in general restricted to trusted sources only. I.e. DE should not generate any byte code for SATML elements which were requested from an untrusted content server. DE shall ignore these elements and any content between the markups or generate an error page.

The SAT Gateway shall be able to reject application containing forbidden functionality. Forbidden functionality is a set of functionality restricted on a application system basis or on an SAT Browser basis. E.g., a restriction of functionality available could be made based on the level of trust of the application system or on the subscription type of the user. A typical usage scenario would be the restriction of the set of allowed proactive commands (e.g. only DISPLAY TEXT, GET INPUT and SELECT ITEM) which the DE accepts via the <sat-gen-stk> interface of SATML.

Further refinements of this security scheme (e.g. look-up tables for public / private SATML elements and attributes) are subject to DE implementation.

The support of security checks of any kind is an optional feature.

4.9. Error Handling

If a page is parsed in the gateway

- which is not well formed,
- where security requirements are not fulfilled,
- where WML / SATML syntax errors are encountered,
- or any other reason prevents successful processing,

the gateway shall send to the browser an encoded SATML page with an error message. WML / SATML syntax errors like

- the invalid usage of variables,
- invalid attribute types,
- an odd length of hexadecimal strings

shall be handled in the same way as XML syntax errors, i.e. like not well formed documents.

5. SATML / SBC Conversion Guidelines

5.1. Unknown DTD

SATML / WML decks encoded with alternate DTD may include elements or attributes that are not recognised by certain DEs. In this situation, a DE should render the deck as if the unrecognised tags and attributes were not present. The content contained in unrecognised elements should be rendered.

5.2. Deck and Deck-Level Elements

5.2.1. Document Prologue

A valid SATML / WML deck is a valid XML document and therefore must contain an XML declaration and a document type declaration. The document encoding is also given in the prologue of a XML document. It is an error to omit the prologue. However, there is no counterpart in SBC. This structure may be ignored by the DE.

5.2.2. satml

SATML **<satml>**	**SBC** **DECK**	**Comment**
xml:lang	no counterpart in SBC	May be ignored by DE.
	DeckTag attributes	
	DCS Attribute *SMS default alphabet* *UCS2*	To be determined by DE.
sat-storage *dynamic* *static*	Storage Dynamic/Static *Dynamic* *Static*	
	Deck Identification element	
	Deck Identification Tag	see [SBC 1.0]
	Deck Identification Value Length	To be determined by DE.
	Deck Identification Value	Unique identification of deck, i.e. coded deck URL which must not contain '#' = 0x23 in one byte (because '#' is used as separator between deck and card names). To be determined by GW from http-request to content server. It is assumed that the DE transforms LongDeckName references to CodedDeckName references by efficient means.
	Service Permanent Store Reference	
	SPS Tag	see [SBC 1.0]
	Length	To be determined by DE.
serv-id	SPS value	To identify all decks which belong to a specific service.
	Cleanup Variable List Element (when exiting the deck)	
	CleanUpVarListTag	see [SBC 1.0]
	Length	To be determined by DE.
sat-do-clr attribute of <sat-var> element	VarRefList	Enumerated list of variables references that need to be cleared at the exit of a deck. To be determined by sat-do-

			clr attribute of <sat-var> elements which may appear inside of a <sat-deck> element. If sat-do-clr = true, variable reference shall be added to VarRefList.
		Text Element Table	
		Text Element Table Tag	see [SBC 1.0]
		Length of Text Element Table	To be determined by DE.
<sat-const>		LV value of each element	Text elements are constant texts that cannot be modified via byte code. To be determined by <sat-const> elements which may appear inside of a <sat-deck> element or by DE optimisation processes parsing for example <p> or <select> elements.
		Template TLV	
<template		Template TLV	See <template> element.
		Card TLV	
<card>		Card TLV	See <card> element.
%id-attrs	id	no counterpart in SBC	May be ignored by DE.
	class	no counterpart in SBC	May be ignored by DE.

5.2.3. wml

See satml. The wml element is part of DTD to be compatible with WML.

5.2.4. head

The head element contains deck-level administrative information. No counterpart in SBC. May be ignored by DE.

5.2.5. access

The access element specifies access control information for the entire deck. If a deck does not include an access element, access control is disabled. When access control is disabled, cards in any deck can access this deck. No counterpart in SBC. May be ignored by DE.

5.2.6. meta

The meta element contains generic meta-information relating to the SATML deck. Meta-information is specified with property names and values. This specification does not define any properties, nor does it define how the GW must interpret meta-data. DE are not required to support the meta-data mechanism. No counterpart in SBC.

5.2.7. template

The template element declares a template for cards in a deck. Event bindings specified in the template element apply to all cards in a deck. Specifying an event binding in the template element is equivalent to specifying it in every card

element. A card element may override the behaviour specified in the template element. See `satml` - Template TLV for more information.

SATML `<template`	SBC `Template TLV`	Comment
	CardTemplateTag	see [SBC1.0]
	Length	To be determined by DE.
	Byte code (BC) Element(s) of Template	

	Byte code elements	Allowed byte codes in the card template must not have user interaction or flow control. Therefore the use of the following SBC commands is not allowed within a template:
		• `INIT_VARIABLE_SELECTED`
		• `GO_BACK`
		• `GO_SELECTED`
		• `SWITCH_CASE_ON_VARIABLE`
		• `EXIT`
		• `STK_GENERIC_MACRO`, if the corresponding proactive command allows to launch a contextual menu by pressing the Cancel, Back or Help key:
		  • `GET INKEY`
		  • `GET INPUT`
		  • `SELECT ITEM`
		  • `DISPLAY TEXT`
		  • `PLAY TONE`
		  • `SET UP CALL`
		  • `SEND DTMF`
		
		Byte code templates may be used to manage entries in contextual menus by using the `MANAGE_CONTEXTUAL_MENUITEM` macro.
		
		General Sequence: Byte code Tag (M), Length (M), Attribute Bytes (O), Byte code Structure associated to the tag (M).

5.2.8. card

SATML `<card>`	SBC `CARD`	Comment
xml:lang	no counterpart in SBC	May be ignored by DE.
title	no counterpart in SBC	May be ignored by DE.
ordered	no counterpart in SBC	May be ignored by DE.

		Card Tag Attributes	
newcontext *true* *false*		ResetVar Attribute *reset the set of temporary variables when entering the card* *keep variables context*	
sat-history *true* *false*		DoNotHistorize *1 = do not historize* *0 = historize*	Usually a card shall be inserted in history stack on exit. However, virtual cards introduced by DE shall not be historized. The creation of virtual cards by DE is implementation dependent.
		DoNotUseTemplate	DoNotUseTemplate = false. DoNotUseTemplate = true is RFU.
		ChainNextCard	No counterpart in WML / SATML. Determines default behaviour at the end of a card. GW implementation dependent. Attribute may be used to manage behaviour at the end of a card, if no <do type=… > is present at SATML source code level.
		Card ID Element	
		Card ID Tag	see [SBC 1.0]
		Length	To be determined by DE
%id-attrs	id	Card ID in the deck	To identify uniquely a card within a deck. Card IDs must not be encoded to short names. Ie the transformation of long names to short names applies *only* to deck identification values.
	class	no counterpart in SBC	May be ignored by DE.
		Byte code (BC) Element(s)	
		Byte code elements	To be determined by elements within the <card> element (<!ELEMENT card (onevent*, timer?, (do | p)*>). General Sequence: Byte code Tag (M), Length (M), Attribute Bytes (O), Byte code Structure associated to the tag (M).
%cardev-attrs	onenterforward	no counterpart in SBC	May be ignored by DE.
	onenterbackward	no counterpart in SBC	May be ignored by DE.
	ontimer	no counterpart in SBC	May be ignored by DE.

5.3. Timer Elements

5.3.1. timer

The `timer` element declares a card timer, which exposes a means of processing inactivity or idle time. No counterpart in SBC. May be ignored by DE.

5.4. Control Elements

5.4.1. input

SATML **<input>**	**SBC** **STK_GENERIC_MACRO**$_{GET\ INPUT}$	**Comment**
xml:lang	no counterpart in SBC	May be ignored by DE.
tabindex	no counterpart in SBC	May be ignored by DE.
size	no counterpart in SBC	May be ignored by DE.
	Byte code Attributes	
	LV EncapsulatedRequired	To be determined by DE dependent on expected TERMINAL RESPONSE according to [GSM 11.14]. For the case of a GET INPUT command: EncapsulatedRequired = 0
	Command Type Value	
	Cmdtype	Cmdtype = 0x23 (GET INPUT)
	Command Qualifier Value	
format *N *M	Bit 1 *0 = digits (0-9, *, # and +) only* *1 = alphabet set*	Other formats than *N or *M shall be converted by DE to *M.
	Bit 2 *0 = SMS default alphabet* *1 = UCS2 alphabet*	DCS shall be inherited from deck.
type *text* *password*	Bit 3 *0 = ME may echo user input on the display* *1 = user input shall not be revealed*	
	Bit 4	Cmdqual_Bit4=0 User input shall always be in unpacked format.
	Bit 8	Cmdqual_Bit8=1 Help shall always be managed by contextual menus.
	Destination Device	
	DestDev	DestDev=0x82 (ME)
	Simple TLVs for the STK command	

		Text String	
		Text string tag	see [GSM 11.14]
		Data coding scheme	DCS shall be inherited from deck.
title		Text string	If no title attribute is present, the text preceding the `input` element shall be taken. See implementation notes of chapter 5.5.1 `p - paragraph`.
		Response Length	
		Response length tag	see [GSM 11.14]
emptyok *true* *false*		Minimum length of response *0* *1*	
sat-minlength		Minimum length of response	Specifies the minimum length of input to be accepted. If this attribute is present, the value of the attribute emptyok shall be ignored.
maxlength		Maximum length of response	
		Default Text	
		Default Text string tag	see [GSM 11.14]
		Data coding scheme	DCS shall be inherited from deck.
value		Default Text string	
		Output Variable ID	
name		Value	ID to be determined by DE.
		Help Text Management	
sat-help			No counterpart in SBC of $STK\_GENERIC\_MACRO_{GET\ INPUT}$. However, additional SBC command `SET_HELP` shall be issued by DE **before** issuing the $STK\_GENERIC\_MACRO_{GET\ INPUT}$ if attribute sat-help is present and help string not empty.
%id-attrs	id	no counterpart in SBC	May be ignored by DE.
	class	no counterpart in SBC	May be ignored by DE.

5.4.2. select

There is no unique counterpart in SBC to translate SATML `select` elements and nested elements like `optgroup` and `option`. However, SBC offers the macros INIT_VARIABLE_SELECTED, SWITCH_CASE_ON_VARIABLE, GO_SELECTED and $STK\_GENERIC\_MACRO_{SELECT\ ITEM}$ to perform SATML / SBC conversions. The command sequence generated by DE shall be implementation dependent for the translation of `select` structures.

To ensure interoperability, the following rules have to be kept:

- PCDATA sections between option elements become item data objects, i.e. list entries.
- The order of options shall not be changed by DE.
- Help text of select element´s sat-help attribute shall act as a master help for each item data object, which may be overridden individually by an option element´s sat-help attribute.
- If optional name attribute of select element is present, a value dependent on the user choice has to be assigned by DE to the declared variable. According to WML the variable must be set to the string value of the chosen option element, which is specified with the value attribute.
- If optional onpick attribute is present, a go operation to the specified URL has to be performed (ie corresponding SBC command sequence has to be generated by DE).

For example DE shall generate from the SATML source

```
<select name="X" title="Choose your favourite animal:">
   <option> Dog </option>
   <option> Cat </option>
   <option> Squirrel </option>
</select>
```

a list with item data objects 'Dog' and 'Cat' and 'Squirrel':

```
Choose your favourite animal
Dog
Cat
Squirrel
```

The user shall be able to pick an item data object from the list of options and dependent on (optional) attributes of the option elements, an action should be performed (assignment of a value to a variable, go to another URL etc).

Lists which are too long to be sent in one proactive SIM command to the ME may be splitted by DE before

5.4.3. option

This element specifies a single choice option in a select element. Translation to SBC by DE can only be performed in correlation with a select element. See 5.4.2 select for more information.

5.4.4. optgroup

The optgroup element shall allow the user to group related option elements into a hierarchy. There is no direct counterpart in SBC. However, optgroup structures may be transformed to select, (option)+ structures before conversion to SBC or additional SBC commands may be generated by DE to indicate a preferred grouping.

It has to be ensured by DE that variable assignments and go operations of nested select, (option)+ structures will be translated to SBC according to the rules of section 5.4.2 select.

5.4.5. fieldset

The fieldset element allows the grouping of related fields and text. There is no direct counterpart in SBC. However, fieldset structures may be transformed to control elements (see 5.4), text elements (see 5.5) or navigation and events (see 5.6) structures before conversion to SBC. Additional SBC commands may be generated by DE to indicate the desired grouping.

5.5. Text Elements

5.5.1. p - paragraph

Usually text between `p` elements should be translated to one or more STK_GENERIC_MACRO$_{\text{DISPLAY TEXT}}$ command(s). However, exceptions may occur for the case of nested `input` or `select` elements or even more complicated structures allowed by DTD. For example, preceding text of `input` or `select` elements with no title attribute shall be used as title (alpha identifier) for the respective byte code translation by DE.

If text between `p` elements contains a mix of variables and constants, a concatenation of all the variables and constants into another variable may be performed by DE before a STK_GENERIC_MACRO$_{\text{DISPLAY TEXT}}$ may be issued.

Therefore it is up to DE to decide which command sequence has to be generated to translate `p` elements into SBC. However, if a STK_GENERIC_MACRO$_{\text{DISPLAY TEXT}}$ is to be issued by DE, the following translation rules have to be applied:

SATML <p>	**SBC STK_GENERIC_MACRO$_{\text{DISPLAY TEXT}}$**	**Comment**
xml:lang	no counterpart in SBC	May be ignored by DE.
align	no counterpart in SBC	May be ignored by DE.
mode	no counterpart in SBC	May be ignored by DE.
	Byte code Attributes	
	LV EncapsulatedRequired	To be determined by DE dependent on expected TERMINAL RESPONSE according to [GSM 11.14]. For the case of a DISPLAY TEXT command: EncapsulatedRequired = 0
	Command Type Value	
	Cmdtype	Cmdtype = 0x21 (DISPLAY TEXT)
	Command Qualifier Value	
`sat-`prio *low* *high*	Bit 1 *0 = normal priority* *1 = high priority*	
`sat-`auto-clr *true* *false*	Bit 8 *0 = clear message after a delay* *1 = wait for user to clear message*	
	Destination Device	
	DestDev	DestDev=0x02 (Display)
	Simple TLVs for the STK command	
	Text String	
	Text string tag	see [GSM 11.14]
	Data coding scheme	DCS shall be inherited from deck.
PCDATA	Text string	To be determined by DE from `PCDATA` within `p` elements.

%id-attrs	id	no counterpart in SBC	May be ignored by DE.
	class	no counterpart in SBC	May be ignored by DE.

5.5.2. br

The br element establishes the beginning of a new line. DE should do best effort to support br elements. However, SBC translation shall be implementation dependent (e.g. insertion of carriage return and line feed when text is encoded to SMS default alphabet, generation of additional STK_GENERIC_MACRO$_{\text{DISPLAY TEXT}}$ etc.).

5.5.3. em

No counterpart in SBC. May be ignored by DE.

5.5.4. strong

No counterpart in SBC. May be ignored by DE.

5.5.5. i

No counterpart in SBC. May be ignored by DE.

5.5.6. b

No counterpart in SBC. May be ignored by DE.

5.5.7. u

No counterpart in SBC. May be ignored by DE.

5.5.8. big

No counterpart in SBC. May be ignored by DE.

5.5.9. small

No counterpart in SBC. May be ignored by DE.

5.6. Navigation and Events

5.6.1. anchor

There is no unique counterpart in SBC to translate SATML `anchor`. However, SBC offers some macros like GO_SELECTED, STK_GENERIC_MACRO$_{\text{SELECT ITEM}}$ and MANAGE_CONTEXTUAL_MENU_ITEM to perform SATML / SBC conversions. The command sequence generated by DE shall be implementation dependent for the translation of `anchor` elements.

To ensure interoperability, the following rules have to be kept:

- DE must not ignore `anchor` elements,
- DE has to ensure that byte code will be generated which allows the user to follow a link,

- The title attribute identifying the link may be ignored, if #PCDATA element is present.

5.6.2. a

The `a` element is a short form of the anchor element and is bound to a `go` task without variables. Therefore the same rules shall apply for SATML / SBC translation as for the `anchor` element (see 5.6.1).

5.6.3. do

The `do` element provides a general mechanism for the user to act upon the current card, i.e., a card level user interface element. SBC offers 3 logical contextual menus (`Back Menu`, `Abort Menu`, `Help Menu`) to perform such operations.

SATML `<do>`	SBC `MANAGE_CONTEXTUAL_MENU_ITEM`	Comment
xml:lang	no counterpart in SBC	May be ignored by DE.
optional	no counterpart in SBC	If `optional`=*true*, DE may ignore this do element.
name	no counterpart in SBC	The `name` attribute is used to override deck-level `do` elements at card-level (see [WML 1.1]). DE shall use the `name` attribute internally to decide when a `MANAGE_CONTEXTUAL_MENU_ITEM` macro shall be issued.
	Byte code Attributes	
	Remove/Hide	Remove/Hide = 0, ie Add/Display the item specified.
	Contextual Menu Item Identifier	
type *prev* *help* *reset* *accept*	MenuItemId *BackMenu* *HelpMenu* *AbortMenu* implementation dependent (see comment)	If `type`=*accept*, DE behaviour shall be implementation dependent. Although the translation is implementation dependent, `go` or `setvar` elements inside of `do` elements must not be ignored. It is optional to issue a `MANAGE_CONTEXTUAL_MENU_ITEM` command and to map this action to a contextual menu. If type=*unknown*, DE behaviour shall be implementation dependent.
	Couple of contextual menu item text and URL to go, if item was selected	
	Couple Tag	
	Length	
	Inline Value or Variable Reference TLV	
	Inline Value Tag | VarRef Tag	see [SBC 1.0]

		Inline Value Length | VarRef Length	To be determined by DE.
label		Inline Value | VarRef Value	Item text to be displayed in contextual menu.
		URL to go when item was selected	
%task within do element		URL Reference Tag	see [SBC 1.0]
		Length	To be determined by DE
	href attribute of go	Value	Both card to card and card to deck navigation may be performed. However, a virtual/temporary card may be introduced by DE before, if there is a `setvar` element, to set a variable. SBC sequence to be generated is implementation dependent, e.g. `CARD` → `INIT_VARIABLES` → `GO_SELECTED`.
	prev		DE shall insert byte code to jump to the previous card. SBC sequence to be generated is implementation dependent
	noop		This task may shadow an existing template. Otherwise no SBC code shall be generated by DE.
	refresh		The `refresh` element declares a `refresh` task, indicating an update of the user agent context as specified by the `setvar` elements. User-visible side effects of state changes (e.g. a change in the screen display) may occur during the processing of a `refresh` task. If there are `setvar` elements inside of the `refresh` element, a virtual/temporary card may be introduced by DE to perform the setting of the variable. SBC sequence to be generated is implementation dependent.
%id-attrs	id	no counterpart in SBC	May be ignored by DE.
	class	no counterpart in SBC	May be ignored by DE.

5.6.4. onevent

The `onevent` element binds a task to a particular intrinsic event for the immediately enclosing element, ie, specifying an `onevent` element inside an "`XYZ`" element associates an intrinsic event binding with the "`XYZ`" element.

DE shall ignore any `onevent` element specifying a type that does not correspond to a legal intrinsic event for the immediately enclosing element. All types apart from `onpick` may be ignored by DE for SATML / SBC translation.

5.6.5. go

The go element declares a go task, indicating navigation to a URI.

There is no direct counterpart in SBC to translate a go element. However, the TL[A]V structure 'URL Reference' is part of some SBC commands like GO_SELECTED which may be used by DE to encode SATML into SBC.

SATML `<go>`		SBC `URL REFERENCE`	Comment
accept-charset		no counterpart in SBC	May be ignored by DE.
		Byte code Attributes	
method *post* *get*		POST / GET method *POST method will be used* *GET method will be used*	
sendreferer *true* *false*		SendReferer *SendReferer will be sent* *SendReferer will not be sent*	
		ForcedResident	If DE detects 'sim' namespace (e.g. URL href = "sim:/my.SATMLPage", URL must not be encoded to short form and ForcedResident attribute has to be set. For references to permanent decks only long URIs are used.
		Address Reference or VarRef element	
		Address Reference Tag | VarRef Tag	see [SBC 1.0]
		Address Reference Length | VarRef Length	To be determined by DE
href		Address Reference Value | VarRef Value	see section "Referencing Decks and Cards" in SBC specification [SBC1.0]. If the `href` attribute is a mix of constants and variables, DE shall concatenate these constants and / or variables to a new variable and reference this variable with the VarRef element.
		Parameters	
		Parameter element TLV structure | Constant parameter element	These optional TLVs may be used by DE to encode `postfield` elements (see 5.6.10 `postfield`)
%id-attrs	id	no counterpart in SBC	May be ignored by DE.
	class	no counterpart in SBC	May be ignored by DE.

5.6.6. prev

The `prev` element declares a `prev` task, indicating navigation to the previous URI on the history stack. See 5.6.3 `do` for conversion guidelines of the `prev` element.

5.6.7. refresh

The `refresh` element declares a `refresh` task, indicating an update of the user agent context as specified by `setvar` elements. See 5.6.3 `do` for conversion guidelines of the `refresh` element.

5.6.8. noop

The `noop` element specifies that nothing should be done, ie, "no operation". See 5.6.3 `do` for conversion guidelines of the `noop` element.

5.6.9. setvar

The `setvar` element specifies a variable to be set in the current browser context as a side effect of executing a task (one of `go`, `prev`, `refresh`, which may appear inside of `anchor`, `onevent` or `do`). For SATML / SBC translation an additional card has to be generated by DE to embed a SBC `INIT_VARABLES` command.

Some optimisation operations of DE are highly recommended in a way that a minimum number of `INIT_VARIABLES` commands should be generated.

SATML `<setvar>`		**SBC `INIT VARIABLES`**	**Comment**
		Argument (VarId, Value)	
name		Id of the variable to initialise	To be determined by DE. See [SBC 1.0] for differences between temporary and permanent variables.
		VarRef | Inline Value Tag	see [SBC 1.0]
		Length	To be determined by DE.
value		Value	Value to be assigned by DE to this variable.
…		…	…
%id-attrs	id	no counterpart in SBC	May be ignored by DE.
	class	no counterpart in SBC	May be ignored by DE.

5.6.10. postfield

The `postfield` element specifies a field name and value for transmission to an origin server during a URL request. The `postfield` element must be nested in a `go` element. Therefore translation to SBC may be performed using the optional 'Parameter TLV' or Constant Parameter TLV of the 'URL Reference' structure (see 5.6.5 go).

SATML `<postfield>`	**SBC `URL REFERENCE`**	**Comment**

		Parameters TLV	
		Parameter Tag	see [SBC 1.0]
		Parameter Length	To be determined by DE.
value		Variable reference to get value from	
name		Parameter name (text string)	For encoding the text string, the data coding scheme of the deck shall be used.
%id-attrs	id	no counterpart in SBC	May be ignored by DE.
	class	no counterpart in SBC	May be ignored by DE.

5.7. SATML Telephony Elements

Telephony elements from the WTAI library may be used in SATML sources to be compliant with existing applications and services written in WML. However, the use of these library functions is not recommended.

5.7.1. set-up call (voice control library)

WTAI **`set-up call`**	**SBC** **`STK_GENERIC_MACRO`**$_{SET\ UP\ CALL}$	**Comment**
telmode	no counterpart in SBC	May be ignored by DE.
	Byte code Attributes	
	LV EncapsulatedRequired	To be determined by DE dependent on expected TERMINAL RESPONSE according to [GSM 11.14]. For the case of a SET UP CALL command: EncapsulatedRequired = 0
	Command Type Value	
	Cmdtype	Cmdtype = 0x10 (SET UP CALL)
	Command Qualifier Value	
	'00' = *set up call, but only if not currently busy on another call*	
	Destination Device	
	DestDev	DestDev=0x83 (Network)
	Simple TLVs for the STK command	
	Address	
	Address tag	see [GSM 11.14]
	Length	To be determined by DE.
	TON and NPI	To be determined by DE from dialling

		number string. If dialling number is preceded by a '+', TON/NPI = 0x91.
telnumber	Dialling number string	
result		May be ignored by DE.

5.7.2. make call (public library)

Refer to 5.7.1 set-up call (voice control library).

5.7.3. send DTMF (voice control library)

WTAI **send DTMF**	**SBC** **STK_GENERIC_MACRO**$_{SEND\ DTMF\ COMMAND}$	**Comment**
callid	no counterpart in SBC	May be ignored by DE.
	Byte code Attributes	
	LV EncapsulatedRequired	To be determined by DE dependent on expected TERMINAL RESPONSE according to [GSM 11.14]. For the case of a SEND DTMF COMMAND: EncapsulatedRequired = 0
	Command Type Value	
	Cmdtype	Cmdtype = 0x14 (SEND DTMF COMMAND)
	Command Qualifier Value	
	'00' = *RFU*	
	Destination Device	
	DestDev	DestDev=0x83 (Network)
	Simple TLVs for the STK command	
	DTMF String	
	DTMF String tag	see [GSM 11.14]
	Length	To be determined by DE.
dtmfstring	DTMF String	
result		May be ignored by DE.

5.7.4. send DTMF (public library)

Refer to 5.7.3 send DTMF (voice control library).

5.7.5. send USSD (GSM specific library)

WTAI `send USSD`	SBC `STK_GENERIC_MACRO`$_{SEND\ USSD}$	Comment
ussdtype	no counterpart in SBC	May be ignored by DE.
ussdid	no counterpart in SBC	May be ignored by DE.
	Byte code Attributes	
	LV EncapsulatedRequired	To be determined by DE dependent on expected TERMINAL RESPONSE according to [GSM 11.14]. For the case of a SEND USSD: EncapsulatedRequired = 0
	Command Type Value	
	Cmdtype	Cmdtype = 0x12 (SEND USSD)
	Command Qualifier Value	
	'00' = *RFU*	
	Destination Device	
	DestDev	DestDev=0x83 (Network)
	Simple TLVs for the STK command	
	USSD String	
	USSD String tag	see [GSM 11.14]
	Length	To be determined by DE.
ussddcs	Data Coding Scheme	
ussdstring	USSD String	
result		May be ignored by DE.

5.8. SATML STK Extensions

5.8.1. sat-var

The `sat-var` element declares a temporary variable.

See 5.6.9 setvar and 5.2.2 `satml`

5.8.2. sat-const

The `sat-const` element specifies a constant text string.

See 5.2.2 `satml`.

5.8.3. sat-sps

The `sat-sps` element specifies a permanent variable. Permanent variables reside in the service permanent store (SPS).

See 5.6.9 setvar.

5.8.4. sat-play-tone

SATML **`sat-play-tone`**	**SBC** **`STK_GENERIC_MACRO`**$_{PLAY\ TONE}$	**Comment**
	Byte code Attributes	
	LV EncapsulatedRequired	To be determined by DE dependent on expected TERMINAL RESPONSE according to [GSM 11.14]. For the case of a PLAY TONE command: EncapsulatedRequired = 0
	Command Type Value	
	Cmdtype	Cmdtype = 0x20 (PLAY TONE)
	Command Qualifier Value	
	'00' = *RFU*	
	Destination Device	
	DestDev	DestDev=0x03 (Earpiece)
	Simple TLVs for the STK command	
	Alpha Identifier	
	Alpha identifier tag	see [GSM 11.14]
	Length	To be determined by DE.
sat-title	Alpha Identifier	
	Tone	
	Tone tag	see [GSM 11.14]
	Length	To be determined by DE.
sat-tone *dial* *busy* *congestion* *radio-ack* *radio-gone* *error* *call-wait* *ring* *beep* *ack* *nack*	Tone *'01' = Dial tone* *'02' = Called subscriber busy* *'03' = Congestion* *'04' = Radio path acknowledge* *'05' = Radio path not available* *'06' = Error / special information* *'07' = Call waiting tone* *'08' = Ringing tone* *'10' = General beep* *'11' = Positive acknowledgement tone* *'12' = Negative acknowledgement or error*	
	Duration	

		Duration tag	see [GSM 11.14]
		Length	To be determined by DE.
		Time unit *'02' = Tenth of seconds*	
sat-duration		Time interval	
%id-attrs	id	no counterpart in SBC	May be ignored by DE.
	class	no counterpart in SBC	May be ignored by DE.

5.8.5. sat-inkey

SATML **<sat-inkey>**	**SBC** **STK_GENERIC_MACRO**$_{GET\ Inkey}$	**Comment**
xml:lang	no counterpart in SBC	May be ignored by DE.
	Byte code Attributes	
	LV EncapsulatedRequired	To be determined by DE dependent on expected TERMINAL RESPONSE according to [GSM 11.14]. For the case of a GET INKEY command: EncapsulatedRequired = 0
	Command Type Value	
	Cmdtype	Cmdtype = 0x22 (GET INKEY)
	Command Qualifier Value	
sat-format *N* *M* *Y*	Bit 1 *0 = digits (0-9, *, # and +) only* *1 = alphabet set* *arbitrary*	Other formats than N, M or Y shall be converted by DE to M.
	Bit 2 *0 = SMS default alphabet* *1 = UCS2 alphabet*	DCS shall be inherited from deck.
sat-format M N *Y*	Bit 3 *0 = character sets defined by bit 1 and bit 2 are enabled* *0 = character sets defined by bit 1 and bit 2 are enabled* *1 = character sets defined by bit 1 and bit 2 are disabled and the "Yes/No" response is requested*	Other formats than N, M or Y shall be converted by DE to M.
	Bit 8	Cmdqual_Bit8=1 Help shall always be managed by contextual menus.
	Destination Device	

		DestDev	DestDev=0x82 (ME)
		Simple TLVs for the STK command	
		Text String	
		Text string tag	see [GSM 11.14]
		Data coding scheme	DCS shall be inherited from deck.
sat-title		Text string	If no `sat-title` attribute is present, the text preceding the `sat-inkey` element shall be taken. See implementation notes of chapter 5.5.1 `p - paragraph`.
		Output Variable ID	
sat-name		Value	ID to be determined by DE.
		Help Text Management	
sat-help			No counterpart in SBC of STK_GENERIC_MACRO$_{\text{GET INKEY}}$. However, additional SBC command `SET_HELP` shall be issued by DE **before** issuing the STK_GENERIC_MACRO$_{\text{GET INKEY}}$ if attribute `sat-help` is present and help string not empty.
%id-attrs	id	no counterpart in SBC	May be ignored by DE.
	class	no counterpart in SBC	May be ignored by DE.

5.8.6. sat-send-sms

SATML **`sat-send-sms`**	**SBC** **STK_GENERIC_MACRO$_{\text{SEND SHORT MESSAGE}}$**	**Comment**
	Byte code Attributes	
	LV EncapsulatedRequired	To be determined by DE dependent on expected TERMINAL RESPONSE according to [GSM 11.14]. For the case of a SEND SHORT MESSAGE command: EncapsulatedRequired = 0
	Command Type Value	
	Cmdtype	Cmdtype = 0x13 (SEND SHORT MESSAGE)
	Command Qualifier Value	
sat-cmdqual	Command qualifier	

		Destination Device	
		DestDev	DestDev=0x83 (Network)
		Simple TLVs for the STK command	
		Alpha Identifier	
		Alpha identifier tag	see [GSM 11.14]
		Length	To be determined by DE.
sat-title		Alpha Identifier	If no sat-title attribute is present, the text preceding the `sat-send-sms` element shall be taken. See implementation notes of chapter 5.5.1 `p - paragraph`.
		Address	
		Address tag	see [GSM 11.14]
		Length	To be determined by DE.
		TON and NPI	To be determined by DE from `sat-smsc` string. If SMSC number is preceded by a '+', TON/NPI = 0x91.
sat-smsc		Dialling number string	For the case of the SEND SHORT MESSAGE command the dialling number string of the Address TLV holds the destination address of the short message service centre.
		SMS TPDU	
		SMS TPDU tag	see [GSM 11.14]
		Length	To be determined by DE.
sat-dest sat-pid sat-dcs sat-period PCDATA		SMS TPDU Destination address Protocol Identifier Data Coding Scheme Validity Period User data	If dialling number in `sat-dest` is preceded by a '+', TON/NPI = 0x91. The user data of the short message shall be determined by DE from `PCDATA` within `sat-send-sms` elements.
%id-attrs	id	no counterpart in SBC	May be ignored by DE.
	class	no counterpart in SBC	May be ignored by DE.

5.8.7. sat-setup-call

SATML **`<sat-setup-call>`**	**SBC** **`STK_GENERIC_MACRO`**$_{SET\ UP\ CALL}$	**Comment**
	Byte code Attributes	

		LV EncapsulatedRequired	To be determined by DE dependent on expected TERMINAL RESPONSE according to [GSM 11.14]. For the case of a SET UP CALL command: EncapsulatedRequired = 0
		Command Type Value	
		Cmdtype	Cmdtype = 0x10 (SET UP CALL)
		Command Qualifier Value	
sat-cmdqual		Command Qualifier	
		Destination Device	
		DestDev	DestDev=0x83 (Network)
		Simple TLVs for the STK command	
		Alpha Identifier (user confirmation phase)	
sat-confirm		Alpha identifier tag	see [GSM 11.14]
		Length	To be determined by DE.
		Alpha identifier	The alpha identifier must be coded as for EF_{ADN} (see GSM 11.11).
		Address	
sat-dest		Address tag	see [GSM 11.14]
		Length	To be determined by DE.
		TON and NPI	To be determined by DE from dialling number string. If dialling number is preceded by a '+', TON/NPI = 0x91.
		Dialling number string	Dialling number string is coded as for EF_{ADN} (see GSM 11.11).
		Alpha Identifier (call set up phase)	
sat-title		Alpha identifier tag	see [GSM 11.14]
		Length	To be determined by DE.
		Alpha identifier	The alpha identifier must be coded as for EF_{ADN} (see GSM 11.11).
%id-attrs	id	no counterpart in SBC	May be ignored by DE.
	class	no counterpart in SBC	May be ignored by DE.

5.8.8. sat-send-ussd

SATML **<sat-send-ussd>**	**SBC** **STK_GENERIC_MACRO**$_{SEND\ USSD}$	**Comment**

		Byte code Attributes	
		LV EncapsulatedRequired	To be determined by DE dependent on expected TERMINAL RESPONSE according to [GSM 11.14]. For the case of a SEND USSD: EncapsulatedRequired = 0
		Command Type Value	
		Cmdtype	Cmdtype = 0x12 (SEND USSD)
		Command Qualifier Value	
		'00' = *RFU*	
		Destination Device	
		DestDev	DestDev=0x83 (Network)
		Simple TLVs for the STK command	
		Alpha Identifier	
sat-title		Alpha identifier tag	see [GSM 11.14]
		Length	To be determined by DE.
		Alpha identifier	The alpha identifier must be coded as for EF_{ADN} (see GSM 11.11).
		USSD String	
sat-dcs sat-data		USSD String tag	see [GSM 11.14]
		Length	To be determined by DE.
		Data Coding Scheme	The data coding scheme must be coded as for Cell Broadcast defined in GSM 03.38.
		USSD String	The coding of the USSD string is defined in GSM 02.30.
%id-attrs	id	no counterpart in SBC	May be ignored by DE.
	class	no counterpart in SBC	May be ignored by DE.

5.8.9. sat-local-info

SATML **<sat-local-info>**	**SBC** **STK_GENERIC_MACRO**$_{ProvideLocalInformation}$	**Comment**
	Byte code Attributes	
	LV EncapsulatedRequired	To be determined by DE dependent on expected TERMINAL RESPONSE according to [GSM 11.14]. For the case of a PROVIDE LOCAL

			INFORMATION: EncapsulatedRequired = 1, i.e. result of the ME is encapsulated in LV (TLV / TLV). No processing is done as complex structure not known.
		Command Type Value	
		Cmdtype	Cmdtype = 0x26 (PROVIDE LOCAL INFORMATION)
		Command Qualifier Value	
		Command Qualifier	
		Destination Device	
		DestDev	DestDev=0x82 (ME)
		Output Variable Id	
sat-name		Id of the variable where to store the result	
%id-attrs	id	no counterpart in SBC	May be ignored by DE.
	class	no counterpart in SBC	May be ignored by DE.
		SBC **GO_SELECTED**	
		Byte code Attributes	
		TitlePresent = 0	No alpha identifier for the (optional) select item.
		URL reference	
		Tag URL reference	To be determined by DE.
		Length	To be determined by DE
		URL byte code attributes	
sat-method *get* *post*		POST / GET Method *0* *1*	
		SendReferer = 0	
		ForcedResident = 0	
		CardOnly = 0	
		Address reference or VarRef element	
		Address reference / VarRef tag	To be determined by DE.
		Address reference / VarRef length	To be determined by DE.
sat-href		Address reference / VarRef value	
		Parameters	

	Parameter tag	
	Parameter length	
sat-name	Variable reference to get value from	Id of the variable which was calculated before shall be inserted by DE. At runtime this is a reference to the result delivered by the `PROVIDE LOCAL INFORMATION` STK command.
sat-name	Parameter name	Name of the variable. DCS shall be inherited from deck.

5.8.10. sat-refresh

SATML `sat-refresh`		SBC `STK_GENERIC_MACRO`$_{REFRESH}$	Comment
		Byte code Attributes	
		LV EncapsulatedRequired	To be determined by DE dependent on expected TERMINAL RESPONSE according to [GSM 11.14]. For the case of a REFRESH command: EncapsulatedRequired = 0
		Command Type Value	
		Cmdtype	Cmdtype = 0x01 (REFRESH)
		Command Qualifier Value	
sat-cmdqual		Command qualifier	
		Destination Device	
		DestDev	DestDev=0x82 (ME)
		Simple TLVs for the STK command	
		File List	
		File List tag	see [GSM 11.14]
		Length	To be determined by DE.
sat-files		Number of files	Number shall be determined by DE from list entries in `sat-files`.
sat-files		Files	Files shall be determined by DE from list entries in `sat-files`.
%id-attrs	id	no counterpart in SBC	May be ignored by DE.
	class	no counterpart in SBC	May be ignored by DE.

5.8.11. sat-gen-stk

SATML `sat-gen-stk`		SBC `STK_GENERIC_MACRO`	Comment
		Byte code Attributes	
		LV EncapsulatedRequired	To be determined by DE dependent on expected TERMINAL RESPONSE according to [GSM 11.14].
		Command Type Value	
sat-cmdtype		Cmdtype	
		Command Qualifier Value	
sat-cmdqual		Command qualifier	
		Destination Device	
sat-destdevice		DestDev	
		Simple TLVs for the STK command	
sat-data		TLV structures	DE shall perform checks to ensure that `sat-`data contains only correctly coded TLVs as far as such checks are possible at compile time. The browser behaviour will be undefined, if invalid encoded TLVs are sent by the gateway to the browser.
%id-attrs	id	no counterpart in SBC	May be ignored by DE.
	class	no counterpart in SBC	May be ignored by DE.

5.8.12. sat-exit

SATML `<sat-exit>`	SBC `EXIT`	Comment
	Byte code Attributes	
sat-cleanbuf *false* *true*	CleanBuffer *0 = Do not clean the execution buffer* *1 = Clean up execution buffer*	
	Exit List Element	
	VarRefList tag	To be determined by DE.
	Length	To be determined by DE.
sat-outvarlist	Variable ID of 1st variable in list	
sat-outvarlist	…	
sat-outvarlist	Variable ID of nth variable in list	

%id-attrs	id	no counterpart in SBC	May be ignored by DE.
	class	no counterpart in SBC	May be ignored by DE.

5.8.13. sat-encrypt

SATML **<sat-encrypt>**		**SBC** **ENCRYPT**	**Comment**
		Key element reference for encrypt and sign	
sat-check *none* *mac*		SPI, Bits 1 and 0 *00* *10*	
sat-crypt *false* *true*		SPI, Bit 3 *0* *1*	
sat-kic		Kic	
sat-kid		Kid	
		Output value element (result of operations) in SecMsg format	
sat-out		DestVarId variable	
		Variable list reference element to encrypt	
		VarRefList tag	To be determined by DE.
		Length	To be determined by DE.
sat-inlist		Variable ID of 1st variable in list	
sat-inlist		…	
sat-inlist		Variable ID of nth variable in list	
%id-attrs	id	no counterpart in SBC	May be ignored by DE.
	class	no counterpart in SBC	May be ignored by DE.

5.8.14. sat-decrypt

	SBC **Secure Message Structure**	**Comment** **Value of SecMsg structure must be set for SBC DECRYPT**
	SecMsg message identification element	To be determined by DE.
	SecMsg Tag	To be determined by DE.

		Length	To be determined by DE.
		Key identification	
sat-check *none* *mac*		SPI, Bits 1 and 0 *00* *10*	
sat-crypt *false* *true*		SPI, Bit 3 *0* *1*	
sat-kic		Kic	
sat-kid		Kid	
		MAC element	optional, dependent on `sat-check`
sat-mac		Length	To be determined by DE.
		Value	
		Enciphered data block	optional, dependent on `sat-crypt`
sat-in		Length	To be determined by DE.
		Value	
SATML **<sat-decrypt>**		**SBC** **DECRYPT**	**Comment**
		Input data block element in SecMsg format	
		VarRef or InlineValueElement tag	To be determined by DE.
		Length	To be determined by DE.
		Value	Variable ID which is a reference to the previously set SecMsg structure.
		Output variable list reference	
sat-outlist sat-outlist sat-outlist		VarRefList tag	To be determined by DE.
		Length	To be determined by DE.
		Variable ID of 1st variable in list	
		…	
		Variable ID of nth variable in list	
%id-attrs	id	no counterpart in SBC	May be ignored by DE.
	class	no counterpart in SBC	May be ignored by DE.

5.8.15. sat-plug-in

SATML	**SBC**	

<sat-plug-in>		EXECUTE	Comment
		Byte code attributes	
sat-return *true* *false*		Exit *0 = returns to SAT browser after having processed the execute element* *1 = processes the exit event before execute element is launched*	
		Execute element reference	
sat-uid		Id of the execute element (manufacturer + reference)	
		Input list element	
		Input list tag	To be determined by DE.
		Length	To be determined by DE.
		VarRef list tag	To be determined by DE.
		Length	To be determined by DE.
sat-inlist		Content of 1st item in list	
sat-inlist		…	
sat-inlist		Content of nth item in list	
		Output variable list reference	
		VarRefList tag	To be determined by DE.
		Length	To be determined by DE.
sat-outlist		Variable ID of 1st variable in list	
sat-outlist		…	
sat-outlist		Variable ID of nth variable in list	
%id-attrs	id	no counterpart in SBC	May be ignored by DE.
	class	no counterpart in SBC	May be ignored by DE.

5.9. Character Entities

SATML	SBC SMS Default
<!ENTITY quot “"”> <!-- quotation mark -->	0x22
<!ENTITY amp “&”> <!-- ampersand -->	0x26
<!ENTITY apos “'”> <!-- apostrophe -->	0x27
<!ENTITY lt “<”> <!-- less than -->	0x3C

<!ENTITY gt “>“>	<!-- greater than -->	0x3E
<!ENTITY nbsp “ “>	<!-- non-breaking space -->	0x20
<!ENTITY shy ““>	<!-- soft hyphen -->	0x2D

6. Annex A: SATML encoding examples [Informative]

The following sample is informative not normative. It is not mandatory to generate the same SBC encoding from the given SATML mark-up source as shown below to be compliant with the SIMalliance Toolbox specifications. I.e. the published SBC is only one of several possibilities to encode a given SATML deck as far as SBC semantics is concerned.

6.1. Main Menu

File: ref_main.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE satml SYSTEM "satml105.dtd">

<satml>
  <!-- Welcome text -->
  <card newcontext="true" id="we">
    <p>
      Welcome to SIMalliance.
    </p>

    <do type = "accept">
      <go href = "#mm"/>
    </do>
  </card>

  <card newcontext="true" id="mm">
    <p>
      <!-- main menu -->
      SATML Services
      <select>
        <!-- go to channel choice                                    -->
        <option>
            Now on the air
          <onevent type = "onpick">
            <go href = "#cc">
              <setvar name = "time_choice" value = "now"/>
            </go>
          </onevent>
        </option>

        <!-- go to channel choice                                      -->
        <option>
          Tonight
          <onevent type = "onpick">
            <go href = "#cc">
              <setvar name = "time_choice" value = "tonight"/>
            </go>
          </onevent>
        </option>

        <!-- go to weekly game                                       -->
        <option onpick = "ref_game.xml"> Weekly game    </option>
      </select>
    </p>
  </card>


  <!-- channel choice -->
  <card id="cc">
```

```
    <p>
      <select title = "TV $time_choice">

        <!-- local channel -->
        <!-- additional card must be introduced to get the local info-->
        <option onpick = "#pli"> Your local channel </option>

        <!-- channel   -->
        <option>
          &lt;Standard channel&gt;
          <onevent type = "onpick">
            <go href = "ref_proglist.xml">
              <postfield name = "time_choice" value = "$(time_choice)"/>
               <postfield name = "channel_choice" value = "C1"/>
            </go>
          </onevent>
        </option>

      </select>
    </p>
  </card>

  <!-- provide local information -->
  <card id="pli">
    <sat-local-info sat-name = "location"/>

    <do type = "vnd.sat-process">
      <go href = "ref_proglist.xml">
        <postfield name = "time_choice" value = "$(time_choice)"/>
         <postfield name = "channel_choice" value = "$(location)"/>
      </go>
    </do>
  </card>
</satml>
```

Byte code analysis

SBC	Meaning	Comment
01	**Deck tag**	
82 01 A1	Length coded in 3 bytes	
02 0D 2F 72 65 66 11 6d 61 69 6e 2e 78 6d 6c	Deck identification element */ref_main.xml*	
85	**Card tag + attribute presence flag**	
37	Length	
40	Attributes: ResetVar=1 DoNotHistorize=0 (default) DoNotUseTemplate=0 (default) ChainNextCard=0 (default)	
06 02 77	Card identification element	

65	"we"	
2D	**STK Generic**	
1D	Length	
21	Command type = DISPLAY TEXT	
01	Command qualifier	
02	Destination device = display	
8D	**Text string tag**	
18	Length	
04	DCS	
57 65 6C 63 6F 6D 65 20 74 6F 20 53 49 4D 61 6C 6C 69 61 6E 63 65 2E	Welcome to SIMalliance	
29	**Go selected tag**	
11	Length	
11	**Couple tag**	
0F	Length	
0A	**Inline value tag**	
06	Length	
61 63 63 65 70 74	*Accept*	
0D	**URL tag**	
05	Length	
0^E	**Address reference**	
03	Length	
23 6D 6D	Address reference value *"#mm"*	
85	**Card tag + attribute presence flag**	
5F	Length	
40	Attributes: ResetVar=1 DoNotHistorize=0 (default)	

	DoNotUseTemplate=0 (default) ChainNextCard=0 (default)	
06 02 6D 6D	Card identification element "mm"	
29	**Go selected tag**	
58	Length	
0A	**Inline value tag**	
0^E	Length	
53 41 54 4d 4c 20 53 65 72 76 69 63 65 73	*SATML Services*	
11	**Couple tag**	
16	Length	
0A	**Inline value tag**	
0^E	Length	
4^E 6F 77 20 6F 6^E 20 74 68 65 20 61 69 72	*Now on the air*	
0D	**URL tag**	
04	Length	
0^E	**Address reference**	
02	Length	
23 61	Address reference value *"#a"*	
11	**Couple tag**	
0F	Length	
0A	**Inline value tag**	
07	Length	
54 6f 6e 69 67 68 74	*Tonight*	
0D	**URL tag**	
04	Length	

0E	**Address reference**	
02	Length	
23 62	Address reference value *"#b"*	
11	**Couple tag**	
1D	Length	
0A	**Inline value tag**	
0B	Length	
57 65 65 6b 6c 79 20 67 61 6d 65	*Weekly Game*	
0D	**URL tag**	
0E	Length	
0E	**Address reference**	
0C	Length	
72 65 66 5F 67 61 6d 65 2e 78 6d 6c	Address reference value *"ref_game.xml"*	
05	**Card tag**	
81 80	Length coded in 2 bytes	
06 02 63 63	Card identification element "cc"	
24	**Concatenate tag**	
09	Length	
03	Destination variable ID	Title
0A	**Inline value tag**	
03	Length	
54 56 20	*TV*	
08	**Variable Reference tag**	
01	Length	
10	Variable ID	TimeChoice
29	**Go selected tag**	

6F	Length	
08	**Variable Reference tag**	
01	Length	
03	Variable ID	Title
11	**Couple tag**	
1C	Length	
0A	**Inline value tag**	
12	Length	
59 6f 75 72 20 6c 6f 63 61 6c 20 63 68 61 6e 6e 65 6c	*Your local channel*	
0D	**URL tag**	
06	Length	
0E	**Address reference**	
04	Length	
23 70 6c 69	Address reference value *"#pli"*	
11	**Couple tag**	
4C	Length	
0A	**Inline value tag**	
12	Length	
3c 53 74 61 6e 64 61 72 64 20 63 68 61 6e 6e 65 6c 3e	*<standard channel>*	
0D	**URL tag**	
36	Length	
0E	**Address reference**	
10	Length	
72 65 66 11 70 72 6f 67 6c	Address reference value *"ref_proglist.xml"*	

69 73 74 2e 78 6d 6c		
0C	**Parameter tag**	
0C	Length	
10	Variable reference to get value from	Time Choice
74 69 6d 65 5f 63 68 6f 69 63 65	Name *time_choice*	
0F	**Constant parameter tag**	
14	Length	
0A	**Inline value tag (Parameter value)**	
02	Length	
43 31	*C1*	
0A	**Inline value tag (Parameter name)**	
0^E	Length	
63 68 61 6^E 6E 65 6C 5F 63 68 6F 69 63 65	*channel_choice*	
05	**Card tag**	
41	Length	
06 03 70 6c 69	Card identification element "pli"	
AD	**STK Generic**	
05	Length	
40	Attribute byte	
26	Command type = PROVIDE LOCAL INFO	
00	Command qualifier	
82	Destination device = ME	
11	Output variable ID	Location

29	**Go selected tag**	
33	Length	
0D	**URL tag**	
31	Length	
0E	**Address reference**	
10	Length	
72 65 66 5F 70 72 6F 67 6C 69 73 74 2E 78 6D 6C	Address reference value "ref_proglist.xml"	
0C	**Parameter tag**	
0C	Length	
10	Variable reference to get value from	TimeChoice
74 69 6d 65 5f 63 68 6f 69 63 65	Name *time_choice*	
0C	**Parameter tag**	
0F	Length	
11	Variable reference to get value from	Location
63 68 61 6e 6e 65 6c 5f 63 68 6f 69 63 65	Name *channel_choice*	
85	**Card tag + attribute presence flag**	
15	Length	
20	Attributes: ResetVar=0 (default) DoNotHistorize=1 DoNotUseTemplate=0 (default) ChainNextCard=0 (default)	
06 01 61	Card identification element "a"	
20	**Initialise variable**	

06	Length	
10	ID of variable to initialise	
0A	**Inline value tag**	
03	Length	
6e 6f 77	*now*	
29	**Go selected tag**	
07	Length	
0D	**URL tag**	
05	Length	
0E	**Address reference**	
03	Length	
23 63 63	Address reference value *"#cc"*	
85	**Card tag + attribute presence flag**	
19	Length	
20	Attributes: ResetVar=0 (default) DoNotHistorize=1 DoNotUseTemplate=0 (default) ChainNextCard=0 (default)	
06 01 62	Card identification element "b"	
20	**Initialise variable**	
0A	Length	
10	ID of variable to initialise	
0A	**Inline value tag**	
07	Length	
74 6f 6e 69 67 68 74	*tonight*	
29	**Go selected tag**	
07	Length	
0D	**URL tag**	
05	Length	

0E	**Address reference**	
03	Length	
23 63 63	Address reference value *"#cc"*	

Variable encodings look-up table

Variable name	Coded name
TimeChoice	10
Title (internal)	03
Location	11

6.2. Program List

File: ref_proglist.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE wml SYSTEM "satml105.dtd">

<!-- It is assumed that this page will be generated
     automatically by a content server dependent on
    the provided values of the variables
       time_choice
     channel_choice
-->

<wml>
   <card newcontext="true">
     <p>
       <select name = "prog" title = "TV program">
         <option value = "FAQ on XML"> 20:00 - 20:45 FAQ on XML </option>
         <option value = "Squirrels">  20:45 - 21:30 Squirrels  </option>
         <option value = "News">       21:30 - 21:45 News       </option>
       </select>

       Would you like additional information?
     </p>

     <do type = "accept">
       <go href = "ref_addproginfo.xml">
         <postfield name = "program" value = "$prog"/>
       </go>
     </do>

     <do type = "prev">
       <go href = "sim:/ref_main.xml#mm"/>
     </do>
   </card>
</wml>
```

Byte code analysis

SBC	Meaning	Comment
01	**Deck tag**	

81 DF	Length coded in 2 bytes	
02 01 80	Deck identification element	
85	**Card tag + attribute presence flag**	
81 D9	Length coded in 2 bytes	
40	Attribute Byte 1 0 0 0 0 0 0 ResetVar Attribute: Reset the set of temporary variables when entering the card	
2C	**Manage contextual menu item**	
1E	Length	
01	Contextual menu item identifier MenuId = 0, i.e. Back Menu System / Application item = 0 Card / Operator item = 0 Menu item Id = 1	Contextual menu item entry will be added to "Back" menu.
11	**Couple Tag**	
1B	Length	
0A	**Inline value tag (contextual menu item text)**	
04	Length	
70 72 65 76	*Prev*	
8D	**URL tag (attribute indicator set)**	
13	Length	
10	Attribute byte ("forced resident" set)	
0E	**Address reference**	
10	Length	
2F 72 65 66 11 6D 61 69 6E 2E 78 6D 6C 23 6D 6D	Address reference value (DeckName for resident deck and prefix sim: removed) */ref_main.xml#mm*	
21	**Init variables selected tag**	
77	Length	

00	Destination variable identifier, temporary variable	Prog
0A	**Inline value tag**	
0A	Length	
54 56 20 70 72 6F 67 72 61 6D	*TV program*	
11	**Couple Tag**	
26	Length	
0A	**Inline value tag (item TLV)**	
18	Length	
32 30 3A 30 30 20 2D 20 32 30 3A 34 35 20 46 41 51 20 6F 6E 20 58 4D 4C	*20:00 - 20:45 FAQ on XML*	
0A	**Inline value tag (value TLV)**	
0A	Length	
46 41 51 20 6F 6E 20 58 4D 4C	*FAQ on XML*	
11	**Couple Tag**	
24	Length	
0A	**Inline value tag (item TLV)**	
17	Length	
32 30 3A 34 35 20 2D 20 32 31 3A 33 30 20 53 71 75 69 72 72 65 6C 73	*20:45 - 21:30 Squirrels*	
0A	**Inline value tag (value TLV)**	
09	Length	

53 71 75 69 72 72 65 6C 73	*Squirrels*	
11	**Couple Tag**	
1A	Length	
0A	**Inline value tag (item TLV)**	
12	Length	
32 31 3A 33 30 20 2D 20 32 31 3A 34 35 20 4E 65 77 73	*21:30 - 21:45 News*	
0A	**Inline value tag (value TLV)**	
04	Length	
4E 65 77 73	*News*	
2D	**STK Generic**	
2C	Length	
21	Command type = DISPLAY TEXT	
80	Command qualifier	
02	Destination device = display	
0D	**Text string tag**	
27	Length	
04	DCS	
57 6F 75 6C 64 20 79 6F 75 20 6C 69 6B 65 20 61 64 64 69 74 69 6F 6E 61 6C 20 69 6E 66 6F 72 6D 61 74 69 6F 6E 3F	*Would you like additional information?*	
29	**Go selected tag**	
0F	Length	

0D	**URL tag**	
0D	Length	
0E	**Address reference**	
01	Length	
81	Address reference value (CodedDeckName) *"0x81"*	
0C	**Parameter tag**	
08	Length	
00	Variable reference to get value from	Prog
70 72 6F 67 72 61 6D	Parameter name (text string) *program*	

URL encodings look-up table

URL	Coded deck name
http://www.gdm.de/ref_proglist.xml	80
http://www.gdm.de/ref_addproginfo.xml	81

Variable encodings look-up table

Variable name	Coded name
prog	00

6.3. Additional Program Information

File: ref_addproginfo.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE satml SYSTEM "satml105.dtd">

<!-- It is assumed that this page will be generated
     automatically by a content server dependent on
   the provided value of the variable prog.
-->
<satml>
  <template>
    <do type = "accept">
      <go href = "#in"/>
    </do>

  </template>
```

```
   <card newcontext="true" id="in">
     <p>
       This is a documentary film about
       <a href = "#sq"> squirrels </a>.<br/>
       The <a href = "#au"> authors </a> have
       done best effort to provide comprehensive
       material on the <a href = "#li" > life </a>
       and <a href = "#ha"> habits </a> of squirrels.
     </p>

     <do type = "accept">
       <go href = "sim:/ref_main.xml"/>
     </do>


   </card>

   <card id="sq">
     <p sat-auto-clr = "true"> <em>Squirrels</em> are shy animals. </p>
   </card>

   <card id="au">
     <p sat-auto-clr = "true"> <i>SIMalliance</i> consortium. </p>
   </card>

   <card id="li">
     <p sat-auto-clr = "true"> Squirrels are living in woods. </p>
   </card>

   <card id="ha">
     <p sat-auto-clr = "true"> Squirrels are collecting nuts. </p>
   </card>

</satml>
```

Byte code analysis

SBC	**Meaning**	**Comment**
01	**Deck tag**	
82 01 DB	Length coded in 3 bytes	
02 01 81	Deck identification element	
07	**Card template tag**	
14	Length	
2C	**Manage Contextual menu item tag**	
12	Length	
01	Contextual menu item identifier	
11	**Couple Tag**	
0F	Length	
0A	**Inline Value Tag**	
06	Length	

61 63 63 65 70 74	“accept”	
0D	**URL reference tag**	
05	Length	
0E	Address reference Tag	
03	Length	
23 69 6E	#in	
85	**Card tag + attribute presence flag**	
82 01 19	Length	
50	Attributes: ResetVar=1 DoNotHistorize=0 (default) DoNotUseTemplate=1 ChainNextCard=0 (default)	
06 02 69 6E	Card identification element "in"	
2D	**STK Generic**	
81 9F	Length	
21	Command type = DISPLAY TEXT	
80	Command qualifier	
02	Destination device = display	
0D	**Text string tag**	
81 99	Length	
04	DCS	
54 68 69 73 20 69 73 20 61 20 64 6F 63 75 6D 65 6E 74 61 72 79 20 66 69 6C 6D 20 61 62 6F 75 74 20 00 73 71 75 69 72 72 65 6C 73 2E 0A 54 68 65	This is a documentary film about @squirrels. *LF*The @authors have done best effort to provide comprehensive material on the @life and @habits of squirrels.	

20 00 61 75 74 68 6F 72 73 20 68 61 76 65 20 64 6F 6E 65 20 62 65 73 74 20 65 66 66 6F 72 74 20 74 6F 20 70 72 6F 76 69 64 65 20 63 6F 6D 70 72 65 68 65 6E 73 69 76 65 20 6D 61 74 65 72 69 61 6C 20 6F 6E 20 74 68 65 20 00 6C 69 66 65 20 61 6E 64 20 00 68 61 62 69 74 73 20 6F 66 20 73 71 75 69 72 72 65 6C 73 2E		
29	**Go selected tag**	
70	Length	
0A	**Inline value tag (Title TLV)**	
0C	Length	
43 68 6F 6F 73 65 20 6C 69 6E 6B 3A	Choose link:	
11	**Couple tag**	
1A	Length	
0A	**Inline value tag (item TLV)**	
06	Length	

61 63 63 65 70 74	*Accept*	
8D	**URL tag (attribute indicator set)**	
10	Length	
10	Attribute byte (“forced resident” set)	
0^{E}	Address reference	
0D	Length	
2F 72 65 66 11 6D 61 69 6E 2E 78 6D 6C	/ref_main.xml	
11	**Couple tag**	
12	Length	
0A	**Inline value tag (item TLV)**	
09	Length	
73 71 75 69 72 72 65 6C 73	*squirrels*	
0D	**URL tag**	
05	Length	
0E	Address reference	
03	Length	
23 73 71	**#sq**	
11	**Couple tag**	
10	Length	
0A	**Inline value tag (item TLV)**	
07	Length	
61 75 74 68 6F 72 73	*authors*	
0D	**URL tag**	
05	Length	
0E	Address reference	

03	Length	
23 61 75	**#au**	
11	**Couple tag**	
0D	Length	
0A	**Inline value tag (item TLV)**	
04	Length	
6C 69 66 65	*life*	
0D	**URL tag**	
05	Length	
0E	Address reference	
03	Length	
23 6C 69	**#li**	
11	**Couple tag**	
0F	Length	
0A	**Inline value tag (item TLV)**	
06	Length	
68 61 62 69 74 73	*habits*	
0D	**URL tag**	
05	Length	
0E	Address reference	
03	Length	
23 68 61	**#ha**	
05	**Card tag**	
26	Length	
06 02 73 71	Card identification element "sq"	
2D	**STK Generic**	
20	Length	
21	Command type = DISPLAY TEXT	
00	Command qualifier	

02	Destination device = display	
0D	**Text string tag**	
1B	Length	
04	DCS	
53 71 75 69 72 72 65 6C 73 20 61 72 65 20 73 68 79 20 61 6E 69 6D 61 6C 73 2E	*Squirrels are shy animals.*	
05	**Card tag**	
23	Length	
06 02 61 75	Card identification element "au"	
2D	**STK Generic**	
1D	Length	
21	Command type = DISPLAY TEXT	
00	Command qualifier	
02	Destination device = display	
0D	**Text string tag**	
18	Length	
04	DCS	
53 49 4D 61 6C 6C 69 61 6E 63 65 20 63 6F 6E 73 6F 72 74 69 75 6D 2E	*SIMalliance consortium*	
05	**Card tag**	
2A	Length	
06 02 6C 69	Card identification element "li"	
2D	**STK Generic**	

24	Length	
21	Command type = DISPLAY TEXT	
00	Command qualifier	
02	Destination device = display	
0D	**Text string tag**	
1F	Length	
04	DCS	
53 71 75 69 72 72 65 6C 73 20 61 72 65 20 6C 69 76 69 6E 67 20 69 6E 20 77 6F 6F 64 73 2^{E}	*Squirrels are living in woods.*	
05	**Card tag**	
2A	Length	
06 02 68 61	Card identification element "ha"	
2D	**STK Generic**	
24	Length	
21	Command type = DISPLAY TEXT	
00	Command qualifier	
02	Destination device = display	
0D	**Text string tag**	
1F	Length	
04	DCS	
53 71 75 69 72 72 65 6C 73 20 61 72 65 20 63 6F 6C 6C 65 63 74 69 6E 67 20 6E 75 74 73 2E	*Squirrels are collecting nuts.*	

URL encodings look-up table

URL	Coded deck name
http://www.gdm.de/ref_addproginfo.xml	81

6.4. Game

File: ref_game.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE satml SYSTEM "satml103.dtd">

<satml sat-help = "Enjoy playing the game">
  <template>
    <do type = "accept">
      <go href = "#g"/>
    </do>

    <do type = "back">
      <go href = "#g"/>
    </do>
  </template>

  <card newcontext="true" id = "g">
    <p>
      <select title = "The Game">
        <option sat-help = "read question" onpick = "#q">
         Question of the week
        </option>

        <option sat-help = "vocal answer" onpick = "#c">
         Have a call
        </option>

        <option sat-help = "post your answer" onpick = "#a">
         Post your secret answer
        </option>
      </select>
    </p>
  </card>

  <card id="q">
    <p>
      Which animal eat nuts?
    </p>
  </card>

  <card id="c">
    <sat-setup-call sat-confirm = "Do you want to set up a call?"
                    sat-title   = "calling..."
              sat-dest    = "+4989123456789" />
  </card>

  <card id="a">
    <p>
      Enter your answer:
      <input name = "var_answer"/>
    </p>
    <do type = "accept">
      <go href = "ref_answer.xml">
        <postfield name = "answer" value = "$var_answser"/>
      </go>
    </do>

  </card>
</satml>
```

Byte code analysis

SBC	Meaning	Comment
01	**Deck tag**	
82 01 76	Length coded in 3 bytes	
02 0C 72 65 66 5F 67 61 6D 65 2^E 78 6D 6C	Deck identification element “ref_game.xml”	
07	**Card template tag**	
3F	Length	
A3	**Set Help tag (attribute indicator set)**	
19	Length	
40	Attribute (“reset help string” set)	
0A	**Inline value tag**	
16	Length	
45 6^E 6A 6F 79 20 70 6C 61 79 69 6^E 67 20 74 68 65 20 67 61 6D 65	“Enjoy playing the game”	
2C	**Manage Contextual menu item tag**	
11	Length	
02	Contextual menu item identifier	
11	**Couple Tag**	
0^E	Length	
0A	**Inline Value Tag**	
06	Length	
61 63 63 65 70 74	“accept”	

0D	**URL reference tag**	
04	Length	
0E	Address reference Tag	
02	Length	
23 67	#g	
2C	**Manage Contextual menu item tag**	
0F	Length	
01	Contextual menu item identifier	
11	**Couple Tag**	
0C	Length	
0A	**Inline Value Tag**	
04	Length	
70 72 65 76	“prev”	
0D	**URL reference tag**	
04	Length	
0E	Address reference Tag	
02	Length	
23 67	#g	
85	**Card tag + attribute presence flag**	
81 8C	Length	
50	Attributes: ResetVar=1 DoNotHistorize=0 (default) DoNotUseTemplate=1 ChainNextCard=0 (default)	
06 01 67	Card identification element ("g")	
23	**Set Help tag**	
2D	Length	
0A	**Inline value tag**	
0D	Length	
72 65 61 64 20 71	“read question”	

75 65 73 74 69 6F 6E		
0A	**Inline value tag**	
0C	Length	
76 6F 63 61 6C 20 61 6E 73 77 65 72	“vocal answer”	
0A	**Inline value tag**	
0E	Length	
73 65 6E 64 20 61 6E 20 61 6E 73 77 65 72	“send an answer”	
29	**Go selected tag**	
57	Length	
0A	**Inline value tag (Title TLV)**	
08	Length	
54 68 65 20 47 61 6D 65	Title (“The Game”)	
11	**Couple tag**	
1C	Length	
0A	**Inline value tag (item TLV)**	
14	Length	
51 75 65 73 74 69 6F 6E 20 6F 66 20 74 68 65 20 77 65 65 6B	*Question of the week*	
0D	**URL tag**	
04	Length	
0E	Address reference	
02	Length	

23 71	"#q"	
11	**Couple tag**	
13	Length	
0A	**Inline value tag (item TLV)**	
0B	Length	
48 61 76 65 20 61 20 63 61 6C 6C	*"Have a call"*	
0D	**URL tag**	
04	Length	
0^E	Address reference	
02	Length	
23 63	"#c"	
11	**Couple tag**	
18	Length	
0A	**Inline value tag (item TLV)**	
10	Length	
70 6F 73 74 20 79 6F 75 72 20 61 6^E 73 77 65 72	*"Post your answer"*	
0D	**URL tag**	
04	Length	
0^E	Address reference	
02	Length	
23 61	"#a"	
05	**Card tag**	
21	Length	
06 01 71	Card identification element "q"	
2D	**STK Generic**	
1C	Length	

21	Command type = DISPLAY TEXT	
00	Command qualifier	
02	Destination device = display	
0D	**Text string tag**	
17	Length	
04	DCS	
57 68 69 63 68 20 61 6E 69 6D 61 6C 20 65 61 74 20 6E 75 74 73 3F	*Which animal eat nuts?*	
05	**Card tag**	
3B	Length	
06 01 63	Card identification element "c"	
2D	**STK Generic**	
36	Length	
10	Command type = SETUP CALL	
00	Command qualifier	
83	Destination device = Network	
05	**Alpha identifier (user confirmation phase)**	
1D	Length	
44 6F 20 79 6F 75 20 77 61 6E 74 20 74 6F 20 73 65 74 20 75 70 20 61 20 63 61 6C 6C 3F	“Do you want to set up a call?”	
06	**Address Tag**	
06	Length	
81	TON/NPI	

40 24 63 85 76	*Number ("0442365867")*	
05	**Alpha identifier (call set up phase)**	
0A	Length	
43 61 6C 6C 69 6E 67 2E 2E 2E	Calling…	
05	**Card tag**	
36	Length	
06 01 61	Card identification element "a"	
2D	**STK Generic**	
1D	Length	
23	Command type = GET INPUT	
01	Command qualifier	
82	Destination device = ME	
0D	**Text String**	
13	Length	
F4	DCS	
45 6E 74 65 72 20 79 6F 75 72 20 61 6E 73 77 65 72 3A	Enter your answer :	
11	**Response Length**	
02	Length	
01	Min	
FF	Max	
01	Destination variable	Var_answer
29	**Go selected tag**	
12	Length	
0D	**URL tag**	
10	Length	

0E	Address reference	
01	Length	
82	Address reference value	
0C	**Parameter tag**	
0B	Length	
01	Variable reference to get value from	Var_answer
76 61 72 5F 61 6E 73 77 65 72	Name *var_answer*	

URL encodings look-up table

URL	**Coded deck name**
http://www.gdm.de/ref_answer.xml	82

Variable encodings look-up table

Variable name	**Coded name**
Var_answer	01