



---

## Committee Draft V1.0, March 2003

# Business-Centric Methodology Specification Appendix B: Linking and Switching Version 1.0

## OASIS BCM Technical Committee Draft Specification

March 07, 2004

### 1 Status of this Document

This document specifies a BCM COMMITTEE SPECIFICATION APPENDIX B: LINKING AND SWITCHING for the Business community.

Distribution of this document is unlimited.

The document formatting is based on the Internet Society's Standard RFC format.

#### *This version:*

<http://www.oasis-open.org/committees/BCM/documents/BCM.AppendixB.LinkingSwitching.2003-08-07.pdf>

#### *Errata for this version:*

<http://www.oasis-open.org/committees/BCM/documents/Errata.shtml>

31  
32  
33  
34  
35

*Previous version:*

None

36    **2 Technical Committee Members**

37    David RR Webber, Consultant

38    Mike Lubash, Independent

39    Bruce Peat, Independent

40    FirstName LastName CompanyName

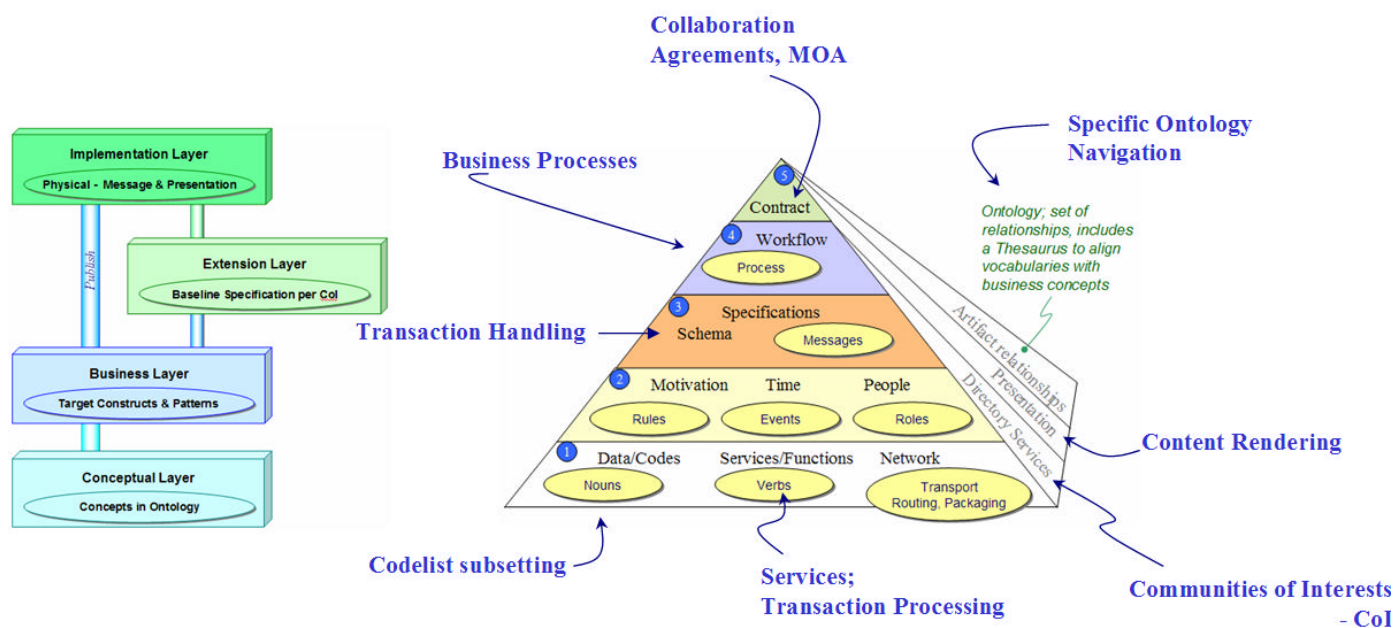
### 41 3 Table of Contents

42	1	Status of this Document.....	1
43	2	Technical Committee Members.....	3
44	3	Table of Contents.....	4
45	4	The Linking and Switching Environment.....	5
46	5	Choice Points – Declarative Context-based Switching.....	8
47	5.1	Choice Point Implementation .....	12
48	5.2	Summary and Next Steps.....	16
49	6	Notices .....	18

## 50 4 The Linking and Switching Environment

51 The focus of the BCM approach is in providing the understanding to allow enterprises to acquire  
 52 and sustain agile information systems that provide reliable business exchanges between  
 53 stakeholders. In analyzing prior legacy approaches and in place systems one key factor is the  
 54 inability to support context driven processes and information exchanges dynamically.  
 55 Particularly in place systems where the logic control is hardwired into program code or locked  
 56 into proprietary delivery systems are inhibitors to agile information exchanges themselves and  
 57 any mitigation or migration techniques seeking to bypass those restrictions. Figure 1 depicts  
 58 some of the context-based switching that occurs at each of the BCM layers within the  
 59 information architecture, along with those which occur at Conceptual, Business, Extension, and  
 60 Implementation layers.

61  
 62 **Figure 1 – Need for Context-based Linking and Switching**



65  
 66  
 67 Today with the advent of individual implementation technologies including XML driven  
 68 software mechanisms, open standards for e-Business transaction formats, and web service aware  
 69 components the challenge is in configuring these to support dynamic context, semantics and  
 70 syntax for interoperable business exchanges. Ironically these same challenges have already  
 71 been architected and tackled previously by agent driven systems designed for dynamic decision  
 72 support. However those prior agent systems suffered from using proprietary interfaces and rule  
 73 bases so that they could not interoperate easily. Instead by using open shared concepts that are  
 74 business-centric and linked to XML formats and exchange mechanisms this shortcoming can be  
 75 addressed (some work has already been done in this direction with efforts such as RuleML and

76 BRML<sup>1</sup>, however these have not focused specifically on the business needs and supporting those  
77 mechanisms directly).

78  
79 The next challenge is ensuring that deployed components actually support the open  
80 specifications mechanisms in a consistent way. Then it becomes possible to create the agile  
81 information exchange systems that users can exploit using a “business-first through choice”  
82 doctrine. This is the focus of the BCM approach, and this section of the BCM specification  
83 details how *Choice Point* mechanisms are needed to enable context driven agile information  
84 exchanges that allow the use of linking and switching across the individual components.

85  
86 Choice Points can be seen as providing three enablers for agile information exchanges:

- 87
- 88 · Context criteria, where the scope of the context extends beyond the local decision point, and  
89 can also require persistence of decisions
- 90
- 91 · Determining context by refining criteria dynamically, and that may include undetermined  
92 start points
- 93
- 94 · Where the context requires a thread manager to establish and track the state of a process.
- 95

96 There are other significant aspects to the implementation of Choice Points, such as consistent  
97 semantic definitions for the context rules and robust process control syntax that allow the user  
98 business requirements to be precisely defined. Those aspects are discussed elsewhere in the  
99 BCM specifications and merely noted where applicable in this section. Also the use of the  
100 Choice Point approach does significantly enhance these other areas, since it is a broad  
101 horizontally applicable technique that can be used to manage all aspects of agile information  
102 exchanges. This serves to highlight the difference with today’s systems that lack Choice Point  
103 technology. Such non-agile systems are therefore static inflexible ‘stovepipe’ solutions that  
104 cannot support dynamic linking and switching and are thus hard to re-purpose and change.

105  
106 A further significant benefit of the Choice Point approach is that it exposes and makes available  
107 the context parameters within a given application layer. This allows business decisions and  
108 choices to be clearly known, classified and selected. Whereas previously applications were built  
109 as a “black box” that could not be easily re-purposed or their suitability to task quickly  
110 determined.

111  
112 Next we consider the implementation constraints. The intention here is to provide a neutral  
113 definition of the BCM Choice Point mechanisms and their XML representations that  
114 implementers can then construct and integrate using popular rule engines. Since each application  
115 own needs will vary it is important that implementers can choose to build just a tailored sub-set  
116 while maintaining interoperability across Choice Points as a prime requirement. This includes  
117 the ability to scale linearly from a simple Choice Point with a single rule-set through to a

---

<sup>1</sup> RuleML – Rules Markup Language and BRML - Business Rules Markup Language and others – complete list is available with links at: <http://xml.coverpages.org/ruleML.html>

118 decision support rule engine operating on a dynamic knowledge base with thousands of facts and  
119 rules.

120

121 In order to implement Choice Point technology requires the ability to manage the inputs (facts)  
122 and outputs (choices) and rule mechanisms applicable to a choice using open consistent formats  
123 in XML and communication protocol standards (see the Choice Point template diagram in figure  
124 2 below). These mechanisms should be “business-first” and accessible to business user  
125 audiences and technical business analysts. This paper details the steps needed in developing this  
126 approach and how that aligns with the overall main body of the BCM specifications.

127

128

129

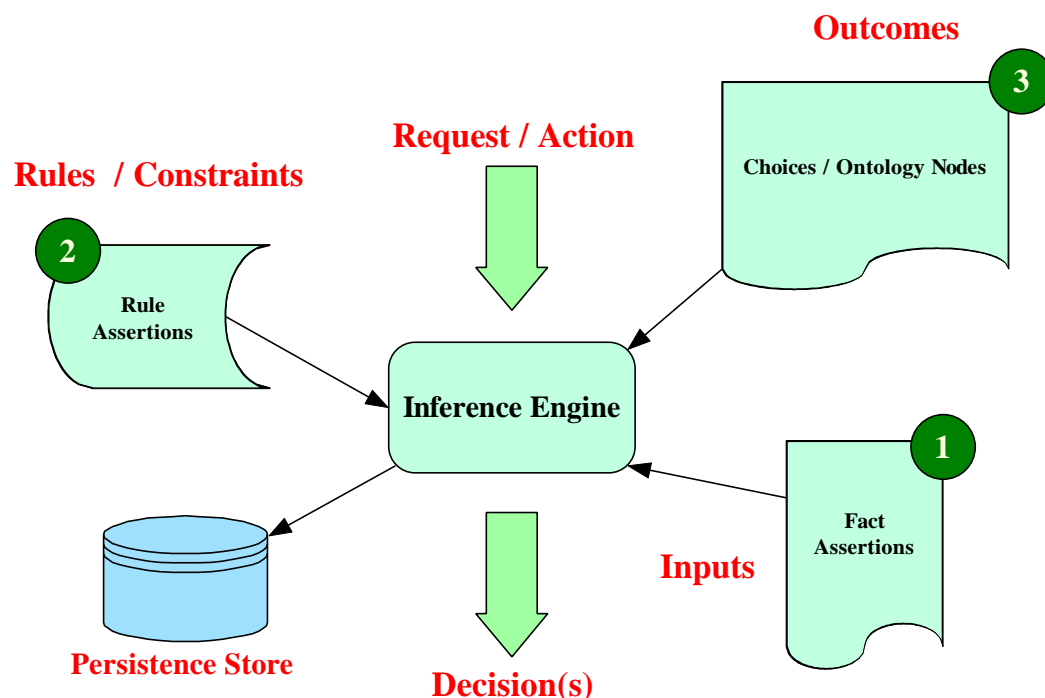
## 130 5 Choice Points – Declarative Context-based Switching

131 The BCM approach emphasizes the need to understand the business problem domain and then  
 132 translate that by layers into physical implementation logic and semantic constructs. Part of that  
 133 process is defining Choice Points within the layers providing the means to capture and  
 134 implement the decision logic. In addition understanding the ontology associated with those  
 135 Choice Points is also required.

136

137

**Figure 2 – Choice Point Conceptual Overview**



138

139

140 As noted in the previous section the Choice Point consists of inputs, business rules and outputs  
 141 that determine the linking and switching to be provided within the business exchange(s). In  
 142 order to configure a Choice Point the business functional needs must be considered and detailed.

143

144 Within the BCM layers<sup>2</sup> there is the need to identify various key interactions and primitive  
 145 entities that describe an interoperable business scenario. These include partner definitions,  
 146 collaborations and roles, process definitions, information transactions and semantic details.  
 147 Using this set of factors and participants we can then state the following:

148

- 149 · Qualifying context is key to ensuring correct relationships between partners in business
- 150 collaborations
- 151 · Knowing context is needed to ensure accurate information capture, packaging and delivery

<sup>2</sup> The diagrams of the BCM layers can be downloaded as large posters from <http://dfas.info>

- 152 · Lack of context control (of the processing and transactions) is the single most prominent  
153 reason why legacy e-Business systems are complex to implement and support  
154 · Providing and managing context is needed to drive dynamic process configuring and control  
155 · Defining ontology both of the Choice Points themselves and including Choice Points within  
156 ontologies (see figure 3 below).

157

158 The context mechanism itself needs to be multifaceted in the types of decision choices that can  
159 be determined and controlled.

160

161 Context can be viewed as a series of cascading Choice Points that have inputs through the  
162 assertion of facts, the operation of rules and constraints, which determine the outcomes from  
163 available choices. These range from the very simple – “if then do” style - to event handlers, to  
164 state management, to complex decision agents that operate on sets of dynamic facts that include  
165 status information about concurrent operations.

166

167 Of course implementations must be able to choose how simple or complex their needs are and  
168 implement Choice Points accordingly. The rules selections may vary from simple binary  
169 choices through to complex decision support questions such as “buy or repair?” logistics. The  
170 BCM Choice Point approach is designed to scale from the simple to the complex in a linear and  
171 consistent way.

172

173 The Choice Point approach lends itself to today's *web service* technology. A Choice Point can  
174 function as a web service, or set of web service calls, that provide dynamic control and decision-  
175 making. Or the Choice Point can be a local component that references assertions and facts from  
176 a web service. Typical uses include tracking and controlling business processes, building  
177 transaction content and providing status of discreet events.

178

179 In examining context to determine the needs it is important to identify that context comes in  
180 many flavours and we can detail the more important types in order that these can be quantified  
181 for a particular implementation. Notice also that context flows through the four layers from the  
182 BCM architecture of conceptual, business, extension and implementation layers.

183

184 Typically the first context that is needed is to determine the Community of Interest (CoI). This  
185 enables one to then exploit re-use by searching within that CoI for components that may be  
186 adapted for the current purpose.

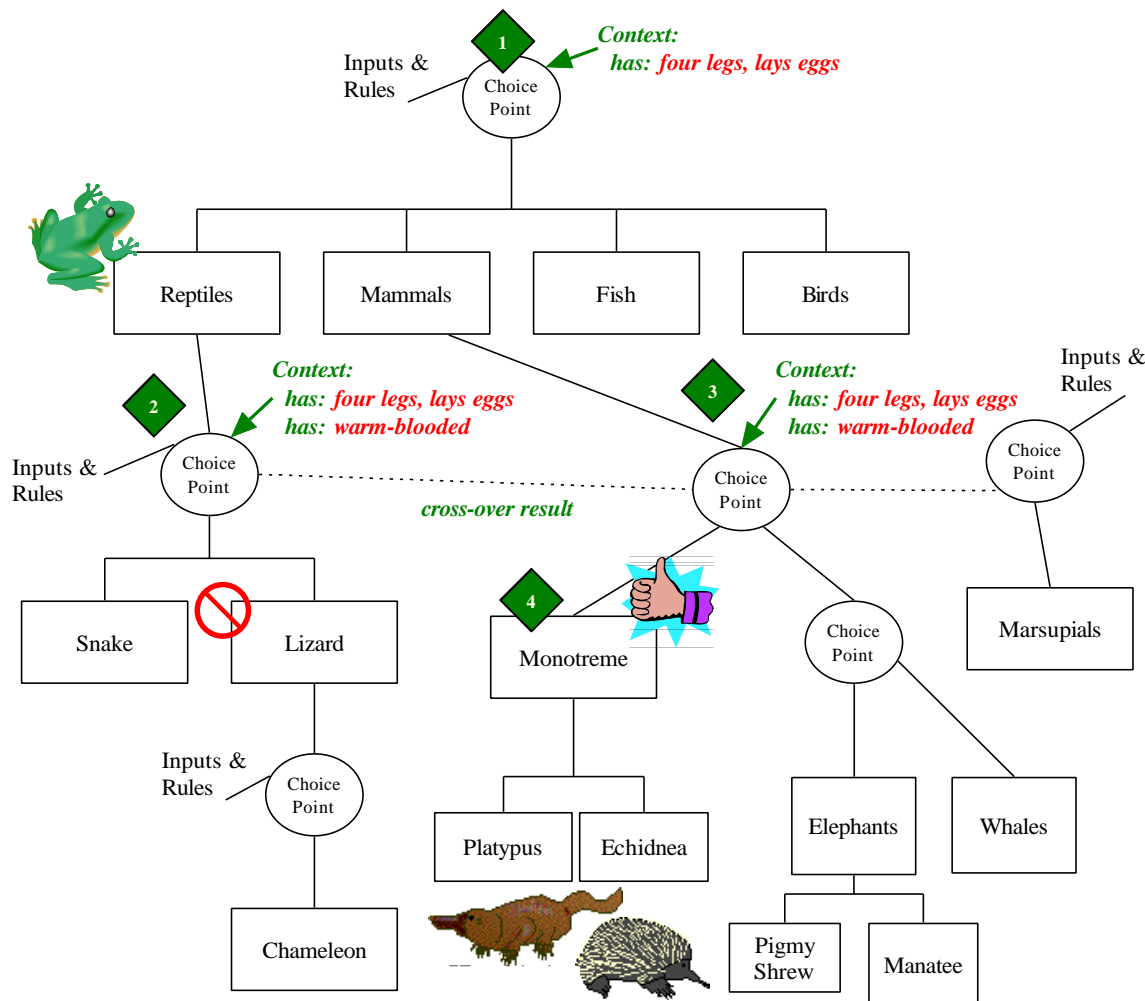
187

188 Next are the business agreement context and the business agreement roles that equate to the  
189 business purpose. Once these are established then the classification of artifacts within that  
190 context can be determined. Classification is a powerful tool for rapidly locating related context  
191 and determining which selection is appropriate from those available. Therefore a classification  
192 hierarchy may contain implicit context switches, or actual Choice Point components (see figure 3

193 for an example of a contextual hierarchy) that can be traversed, and the branching that may occur  
 194 across the hierarchy based on relations and associations<sup>3</sup>.

195  
 196  
 197

**Figure 3 – contextual classification hierarchy with crossovers**



198  
 199  
 200  
 201  
 202  
 203  
 204  
 205  
 206

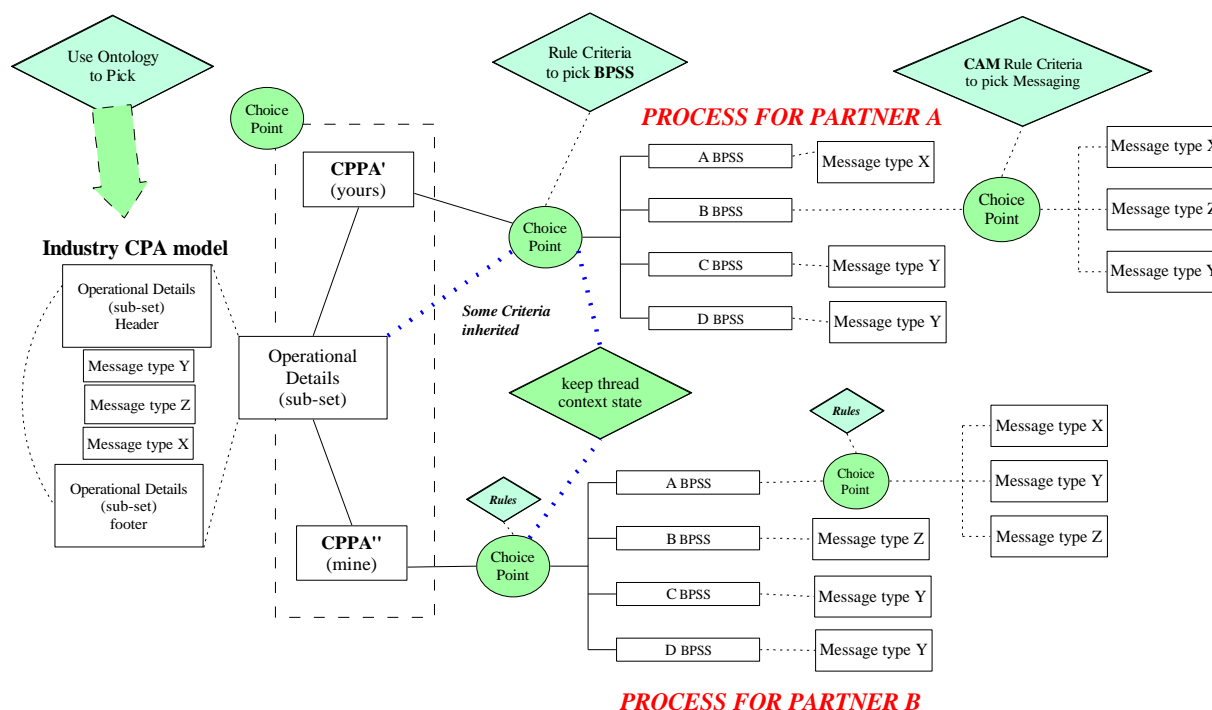
Continuing with the analysis of context types into the implementation layer from which understanding the business process is paramount. This includes process selection context and process tracking context. Below the process is the transaction context followed by the exception-handling context. At the interface to the application systems there is context that is supplied to the decisions and rules that are associated with the information handling.

<sup>3</sup> Note: ebXML registry information model fully supports this use and the 'browse and drilldown' approach.

206 This cascading of Choice Points through the business implementation layers can be seen in  
 207 figure 4, where the ebXML implementation stack<sup>4</sup> is used as an example. The context can be  
 208 summarized as the following:

- 209
- 210 · Community of interest determination (CPPA specification / business ontology)
- 211 · Business agreement context (CPPA specification)
- 212 · Business agreement roles (CPPA specification)
- 213 · Classification of artifacts context (CPPA specification)
- 214 · Process selection context (BPSS specification)
- 215 · Process tracking context (BPSS specification)
- 216 · Transaction context (BPSS specification / CAM specification)
- 217 · Exception handling context (CAM specification)
- 218 · Decisions and rules context (CAM specification)
- 219 · Lookup tables and contextual subsets (CAM specification)

220  
 221  
 222 **Figure 4 – Cascading e-Business choice points within the implementation layer**  
 223



224  
 225  
 226 Reviewing figure 4 from left to right, the initial step is to use the ontology to determine the  
 227 correct community of interest and select the model for the business exchange required. The  
 228 model will include details of the business process and the document exchanges (as shown with

<sup>4</sup> CPPA – Collaboration Partner Profile Agreement (ebXML), BPSS – Business Process Schema Specification (ebXML), CAM – Content Assembly Mechanism (OASIS).

229 the header and footer. Each trading partner then refines these based on their own operational  
230 details, and creates a Choice Point set of inputs, rules and outcomes based on the model. They  
231 then compare these and agree on the specific business process(es) they wish to use, the  
232 transaction messages (their structure format, content semantics and content rules), and update the  
233 context criteria accordingly to enforce these. These actions correspond to determining the  
234 context items summarized in the list immediately above figure 4.

235

236 The thread context state mechanism shown linked between the Choice Points allows both  
237 partners to keep in lock step with each other's business processes as the actual exchanges occur  
238 in their real-world systems (thread management is part of the Choice Point functional  
239 requirements already noted earlier).

240

241 Figure 4 shows a wide variety of possible business process paths and message choices with four  
242 process sequences (A,B,C,D) and three message formats (X, Y, Z). Typically business partners  
243 would pick just a subset of these for their initial implementation needs.

244

245 Choice Points therefore are involved in the entire process; configuring the business partner  
246 collaborations, selecting the details of the business processing, controlling the transaction content  
247 messages and tracking the state of each interchange that occurs.

248

249 As previously noted the Choice Point approach lends itself to today's web service technology as  
250 part of a Service Oriented Architecture (SOA). Each Choice Point can be described using XML  
251 templates formatted as WSDL<sup>5</sup> definitions. So in figure 3, the Choice Points denoted could easily  
252 be implemented as web service driven components that provide control and selection within the  
253 implementation layer.

254

255 The Choice Points could also interact with a registry of definitions so that the complete  
256 behaviour can be externally configured and context driven. With such adaptability this delivers  
257 agile information flows based on business context.

258

## 259 **5.1 Choice Point Implementation**

260 The Choice Points have been described so far as abstract concepts. This section provides design  
261 details of the operation of Choice Points and their behaviors. To understand this we need to first  
262 collect the required Choice Point behaviors discussed so far above and summarize these:

263

- 264 · Allow inputs (facts) to determine outcomes (choices) based on rules
- 265 · Rules can be expressed and asserted non-procedurally with simple business-friendly  
266 constructs and syntax
- 267 · Choice Points can call Choice Points
- 268 · Assertion of facts and / or rules can be passed as inputs to a Choice Point
- 269 · Choice Points may inherit context details

---

<sup>5</sup> Web Service Description Language, a W3C specification for describing web service points, their access and operations.

- 270 · Decisions may be persisted for later process needs
- 271 · Choices can be a simple fixed set, or could be a dynamic set
- 272 · Choice Points are exposed as components of the architecture and not closed as inaccessible
- 273 within a solution
- 274 · Choice Points can communicate via web services and messaging as needed
- 275 · Choice Points can hold the transient state of interactions

276  
277 Next we consider the implementation constraints. The intention here is to provide a neutral  
278 definition of the BCM Choice Point mechanisms and their XML representations that  
279 implementers can then construct and integrate using popular rule engines. Since each application  
280 needs will vary it is important that implementers can choose to build a tailored sub-set while  
281 maintaining interoperability across Choice Points as the prime requirement.

282  
283 Since Choice Points may interact themselves it is vital that the base functionality be established  
284 via the use of an open XML driven service with an API (application programming interface).  
285 Part of establishing this includes the ability to use a broad set of communications via WSDL  
286 definitions. Other OASIS technical specifications have already successfully implemented this  
287 approach, including the OASIS CAM specification. A further implementation need is that the  
288 Choice Point mechanism can be used by other OASIS specifications to provide dynamic context  
289 driven behaviors. Examples that have already been identified include: BPEL, BPSS, CAM,  
290 CPPA, UBL, and the CIQ specifications.

291  
292 In order to construct a consistent XML driven API the following components are needed:

- 293
- 294 · Rule base and consistent decision mechanisms with supporting XML syntax
- 295 · Fact base and consistent representations in XML syntax for context
- 296 · State tracking and ability to assign globally unique thread IDs
- 297 · Query and Response action formats
- 298 · Change action formats
- 299 · Event handling formats
- 300 · Security support with audit trail within the Choice Point implementation

301  
302 This summary is provided here, each of these items is expanded more thoroughly in the Choice  
303 Point technical specification itself<sup>6</sup>. The primary behaviors are listed first, while those behaviors  
304 likely to be optionally included in implementations are listed last.

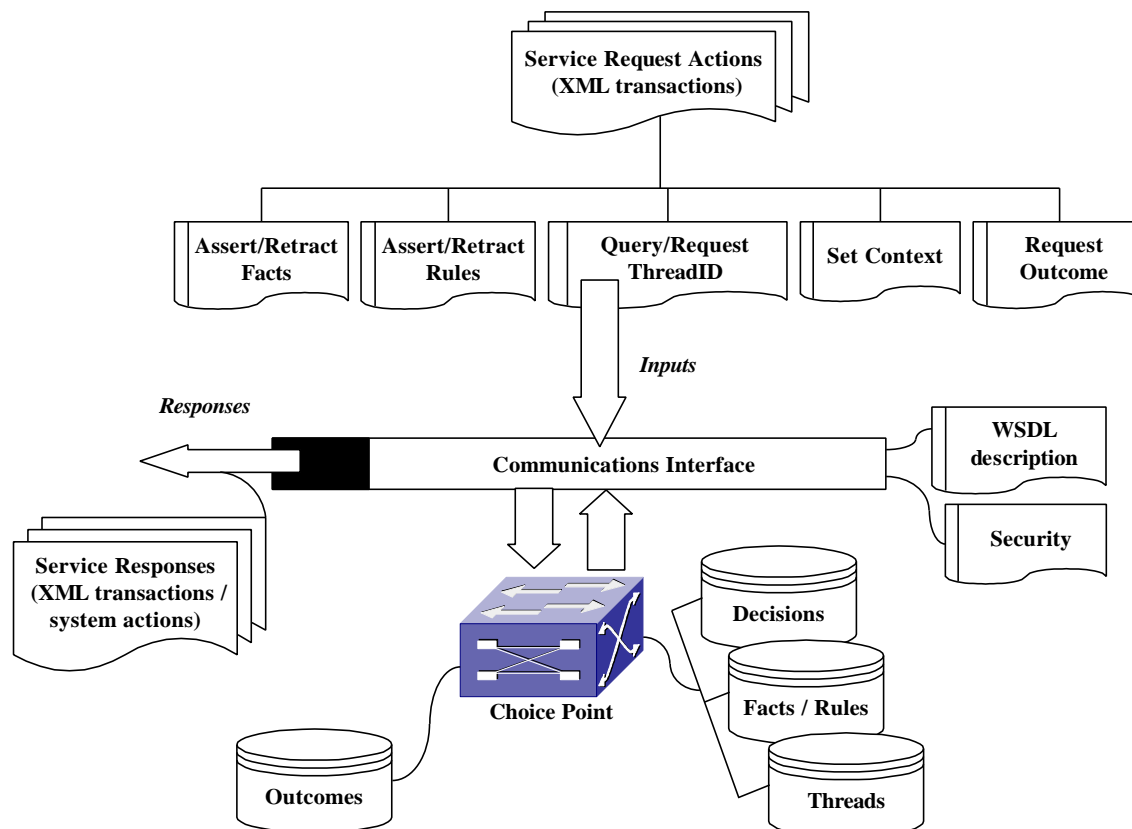
305  
306 Figure 5 depicts these components of the Choice Point implementation.

307  
308

---

<sup>6</sup> See BCM technical specifications for these details.

308 **Figure 5 – Choice Point rule engine implementation components**  
 309



310  
 311  
 312 The Choice Point engine itself can have a variety of behaviors supported by the rule engine. Not  
 313 all may be required, depending on the business application. This flexibility means that the  
 314 Choice Point approach can be implemented directly using popular programming languages,  
 315 without the need for a specialized rule engine, or alternately can be configured to use a rule  
 316 agent. The varieties of anticipated common needs of these behaviors include:

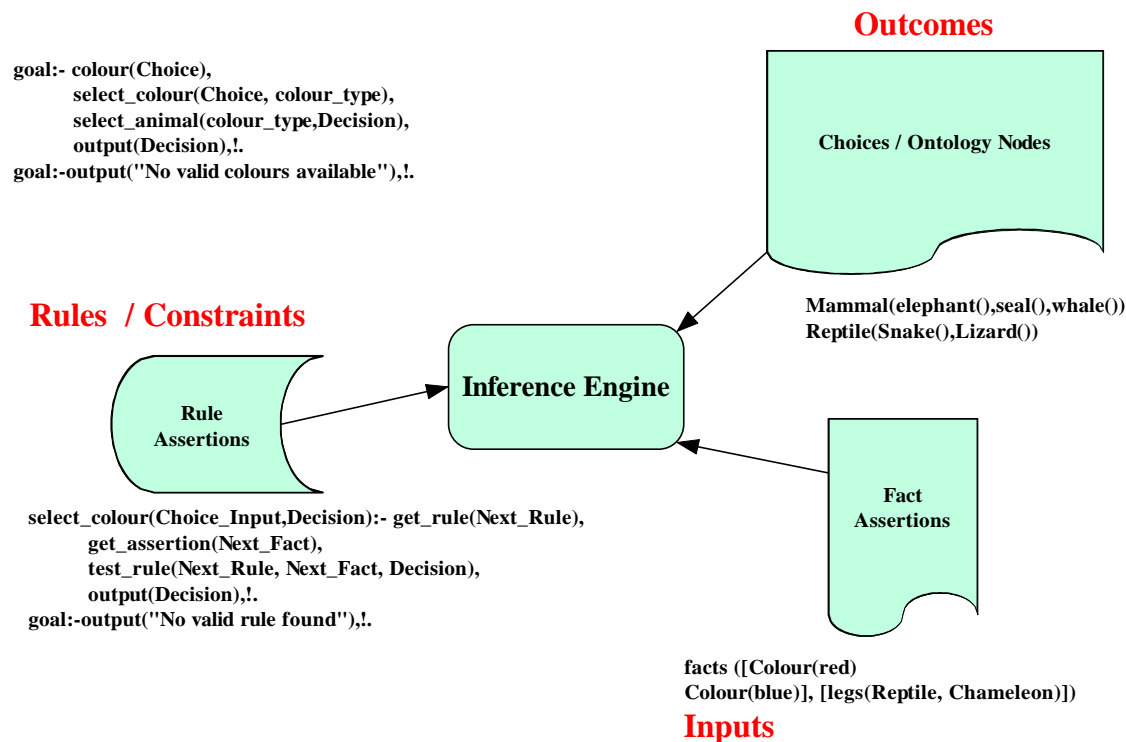
- 317  
 318 · Fact assertion / retraction  
 319 · Rule assertion / retraction  
 320 · State tracking mechanism  
 321 · Simple case rule determination (select-when-otherwise)  
 322 · Solution determination via backtracking supported  
 323 · Solution determination via forward tracking supported  
 324 · Solution determination using constraint logic supported  
 325 · Storage of current state decision memory for later recall (decision threads)  
 326 · Decision testing support (if-then analysis)  
 327 · Audit trail and decision verification (why was this decision chosen?)  
 328 · Event handling support

329

330 To complete this section on Choice Point implementation figure 6 shows a possible  
 331 configuration using a Prolog programming language based inference engine. Prolog has been  
 332 used extensively for decision support implements and a wide variety of proven implementations  
 333 are available. This example is not intended to be normative but merely to show the concepts  
 334 behind implementing dynamic rule based decision processes. These mechanisms then require  
 335 support via the XML formats and syntax of the Choice Point specification. It is therefore helpful  
 336 to understanding those constructs and their behaviors.

337  
 338

**Figure 6 – Example of decision rules processing**



339  
 340

341 Referencing figure 6 above, the interface is shown in the “goal” section that controls the decision  
 342 process. The WSDL interface to the Choice Point will need to expose support for such  
 343 interactions. Similarly the “Rules / Constraints” will be implemented in XML syntax and a  
 344 human friendly front-end provided that allows business users to create these. And then the facts  
 345 and outcomes similarly will be input from a front-end and have XML formats for their creation  
 346 and exchange. The implementer can then provide a bridge between their own internal Prolog  
 347 syntax and the open Choice Point XML formats and syntax. As noted earlier, considerable work  
 348 has already been done in this area of representation of rules logic using XML including such  
 349 work as RuleML – Rules Markup Language and BRML - Business Rules Markup Language and  
 350 others – and a complete list is available with links at: <http://xml.coverpages.org/ruleML.html>.  
 351 Other noteworthy work is that done by the SHOE team – working on Simple HTML Ontology  
 352 Extensions <http://www.cs.umd.edu/projects/plus/SHOE/>. The need is to combine this earlier  
 353 work with the Choice Point requirements to produce an implementation set that can deliver the  
 354 needed behavior overall.

355

## 355 **5.2 Summary and Next Steps**

356 The BCM Choice Point approach provides a vital component for implementing agile information  
357 systems. With the advent of web service based Service Orientated Architectures this component  
358 is urgently required to ensure consistent implementations today. Furthermore the traditional e-  
359 Business systems interfaces within this model also need to transition their processes and content  
360 handling to support Choice Points as a means to deliver interoperability and adaptability.

361  
362 While decision support systems in the past have implemented such techniques they have done so  
363 as closed systems. The opportunities that open rule-formats using XML together with  
364 interoperable communications brings is to remove the limitations of prior architectures and  
365 provide dynamic context driven implementation of enterprise systems.

366  
367 This section of the BCM specifications is intended to facilitate this and form the basis for the  
368 scope of action of the Linking and Switching sub-committee (SC) of the BCM technical  
369 committee (TC).

370  
371 It is anticipated that further liaison and outreach with other OASIS technical committees (TCs)  
372 will occur to refine requirements and the implementation model, and this process has already  
373 begun. Part of the deliverables for the sub-committee will include the creation of W3C WSDL  
374 models for Choice Points that will help other groups to understand the interface from their own  
375 specifications.

376  
377 In parallel with these liaison efforts is the development of an initial Choice Points technical  
378 specification details (the Pareto Principle applies!) leading to prototyping using available rule  
379 engines and a demonstration using selected business scenarios.

380  
381 Those interested in contributing to this work are encouraged to join the OASIS BCM TC and the  
382 Linking and Switching SC, more details on this are available from the OASIS website  
383 (<http://www.oasis-open.org>).



384 **6 Notices**

385 *OASIS takes no position regarding the validity or scope of any intellectual property or other*  
386 *rights that might be claimed to pertain to the implementation or use of the technology described*  
387 *in this document or the extent to which any license under such rights might or might not be*  
388 *available; neither does it represent that it has made any effort to identify any such rights.*  
389 *Information on OASIS's procedures with respect to rights in OASIS specifications can be found*  
390 *at the OASIS website. Copies of claims of rights made available for publication and any*  
391 *assurances of licenses to be made available, or the result of an attempt made to obtain a general*  
392 *license or permission for the use of such proprietary rights by implementers or users of this*  
393 *specification, can be obtained from the OASIS Executive Director. OASIS invites any interested*  
394 *party to bring to its attention any copyrights, patents or patent applications, or other proprietary*  
395 *rights which may cover technology that may be required to implement this specification. Please*  
396 *address the information to the OASIS Executive Director.*

397 **Copyright © OASIS Open 2003 / 2004. All Rights Reserved.** *This document and translations*  
398 *of it may be copied and furnished to others, and derivative works that comment on or otherwise*  
399 *explain it or assist in its implementation may be prepared, copied, published and distributed, in*  
400 *whole or in part, without restriction of any kind, provided that the above copyright notice and*  
401 *this paragraph are included on all such copies and derivative works. However, this document*  
402 *itself may not be modified in any way, such as by removing the copyright notice or references to*  
403 *OASIS, except as needed for the purpose of developing OASIS specifications, in which case the*  
404 *procedures for copyrights defined in the OASIS Intellectual Property Rights document must be*  
405 *followed, or as required to translate it into languages other than English. The limited*  
406 *permissions granted above are perpetual and will not be revoked by OASIS or its successors or*  
407 *assigns.*

408 This document and the information contained herein is provided on an "AS IS" basis and OASIS  
409 DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT  
410 LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN  
411 WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF  
412 MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE."

413 Where, pursuant to a notification under this Policy, the OASIS Board of Directors is aware at the  
414 time of publication of proprietary rights claimed with respect to an OASIS specification, or the  
415 technology described or referenced therein, such specification shall contain the following notice:

416 OASIS has been notified of intellectual property rights claimed in regard to some or all of the  
417 contents of this specification. For more information consult the online list of claimed rights.

418 This document and the information contained herein is provided on an "AS IS" basis and OASIS  
419 DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT  
420 LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN  
421 WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF  
422 MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.