

1
2
3 Business-Centric Methodology Specification
4 Appendix B: *Linking and Switching*
5 *with Context and Choice Points*
6 Version 0.01

7
8 OASIS BCM Technical Committee

9
10 August 07, 2003

11
12
13 1 Status of this Document

14
15 This document specifies a BCM SPECIFICATION APPENDIX B: LINKING AND
16 SWITCHING, WITH CONTEXT AND CHOICE POINTS for the Business community.

17
18 Distribution of this document is unlimited.

19
20 The document formatting is based on the Internet Society's Standard RFC format.

21
22 ***This version:***

23
24 [http://www.oasis-
open.org/committees/BCM/documents/BCM.AppendixB.LinkinSwitching.2003-08-07.pdf](http://www.oasis-
25 open.org/committees/BCM/documents/BCM.AppendixB.LinkinSwitching.2003-08-07.pdf)

26
27 ***Errata for this version:***

28
29 <http://www.oasis-open.org/committees/BCM/documents/Errata.shtml>

30
31 ***Previous version:***

32
33 [None](#)

35 2 Technical Committee Members

36 David RR Webber, Consultant

37 Mike Lubash, Independent

38 Bruce Peat, Independent

39 FirstName LastName CompanyName

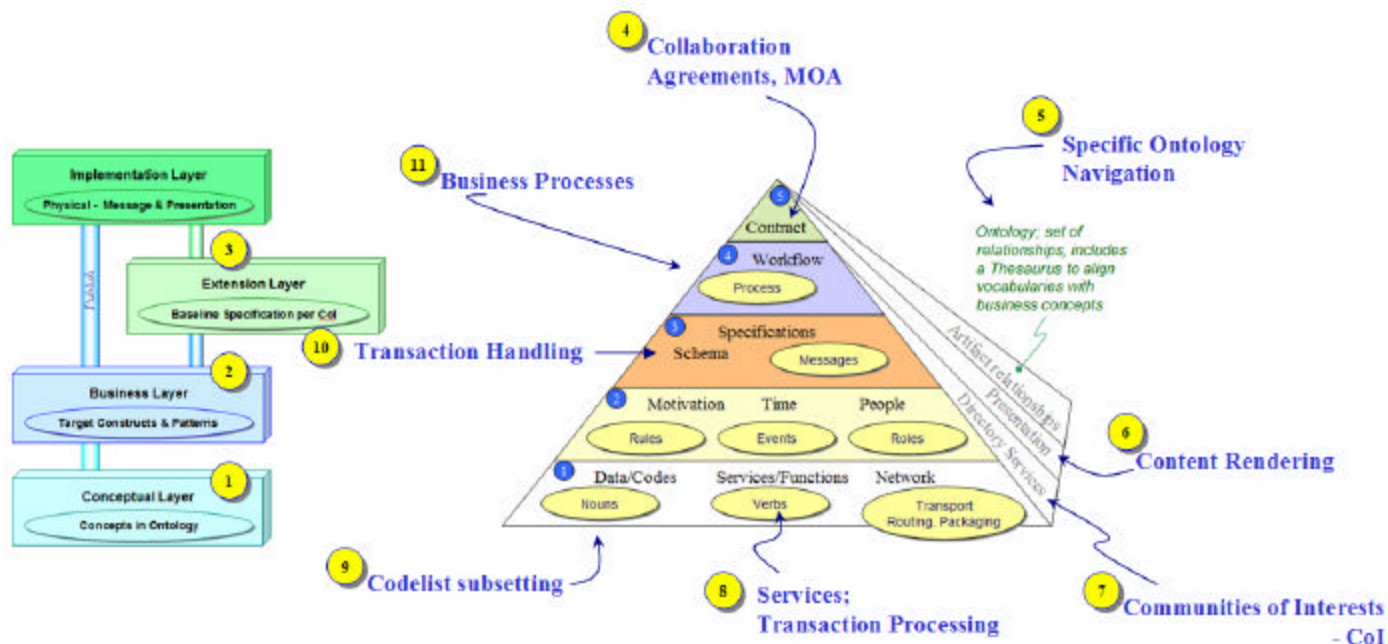
40 3 Table of Contents

41	1	Status of this Document.....	1
42	2	Technical Committee Members	2
43	3	Table of Contents.....	3
44	4	The Linking and Switching Environment.....	4
45	5	Choice Points – Declarative Context -based Switching.....	7
46		5.2 Choice Point Implementation	11
47		5.3 Summary and Next Steps	15

48 **4 The Linking and Switching Environment**

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

59 **4.1.1.1 Figure 1 – Need for Context-based Linking and Switching**



61
62
63
64
65
66
67
68
69
70
71
72

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

73 BRML¹, however these have not focused specifically on the business needs and supporting those
74 mechanisms directly).

75
76 The next challenge is ensuring that deployed components actually support the open
77 specifications mechanisms in a consistent way. Then it becomes possible to create the agile
78 information exchange systems that users can exploit using a “business-first through choice”
79 doctrine. This is the focus of the BCM approach, and this section of the BCM specification
80 details how *Choice Point* mechanisms are needed to enable context driven agile information
81 exchanges that allow the use of linking and switching across the individual components.

82
83 Choice Points can be seen as providing three enablers for agile information exchanges:

- 84
- 85 · Context criteria, where the scope of the context extends beyond the local decision point, and
86 can also require persistence of decisions
- 87
- 88 · Determining context by refining criteria dynamically, and that may include undetermined
89 start points
- 90
- 91 · Where the context requires a thread manager to establish and track the state of a process.
- 92

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

102
103 A further significant benefit of the Choice Point approach is that it exposes and makes available
104 the context parameters within a given application layer. This allows business decisions and
105 choices to be clearly known, classified and selected. Whereas previously applications were built
106 as a “black box” that could not be easily re-purposed or their suitability to task quickly
107 determined.

108
109 Next we consider the implementation constraints. The intention here is to provide a neutral
110 definition of the BCM Choice Point mechanisms and their XML representations that
111 implementers can then construct and integrate using popular rule engines. Since each application
112 own needs will vary it is important that implementers can choose to build just a tailored sub-set
113 while maintaining interoperability across Choice Points as a prime requirement. This includes
114 the ability to scale linearly from a simple Choice Point with a single rule-set through to a

¹ 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>

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

117
118 In order to implement Choice Point technology requires the ability to manage the inputs (facts)
119 and outputs (choices) and rule mechanisms applicable to a choice using open consistent formats
120 in XML and communication protocol standards (see the Choice Point template diagram in figure
121 2 below). These mechanisms should be “business-first” and accessible to business user
122 audiences and technical business analysts. This paper details the steps needed in developing this
123 approach and how that aligns with the overall main body of the BCM specifications.

124

125

126

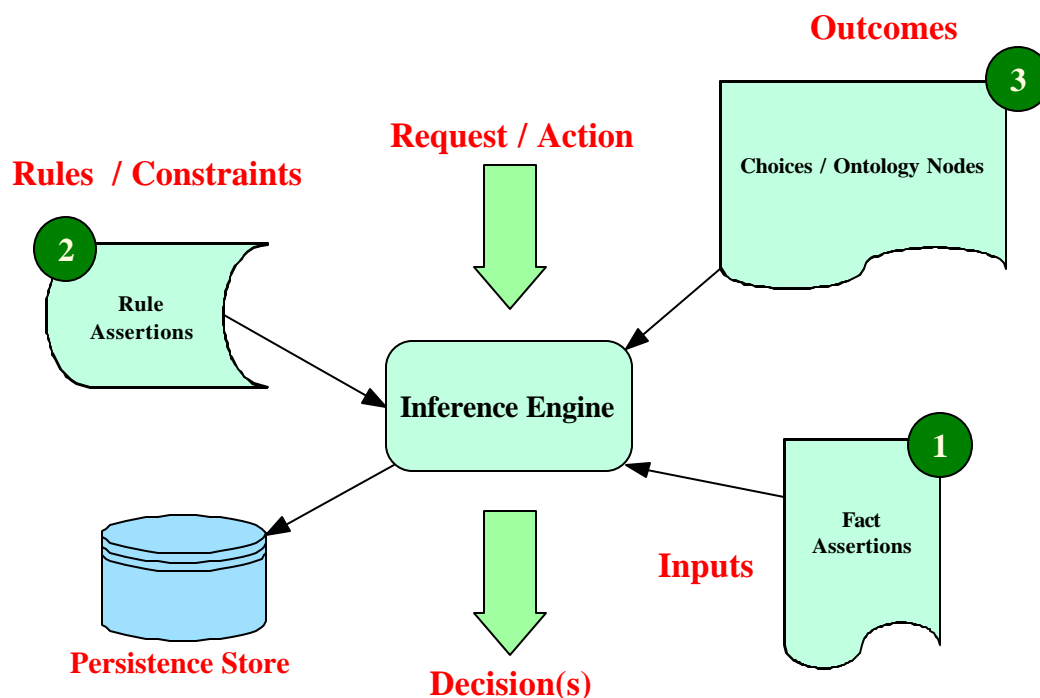
127 5 Choice Points – Declarative Context-based Switching

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

133

134

5.1.1.1 Figure 2 – Choice Point Conceptual Overview



135

136

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

140

141 Within the BCM layers² there is the need to identify various key interactions and primitive
 142 entities that describe an interoperable business scenario. These include partner definitions,
 143 collaborations and roles, process definitions, information transactions and semantic details.
 144 Using this set of factors and participants we can then state the following:

145

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

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

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

154

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

157

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

163

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

169

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

175

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

180

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

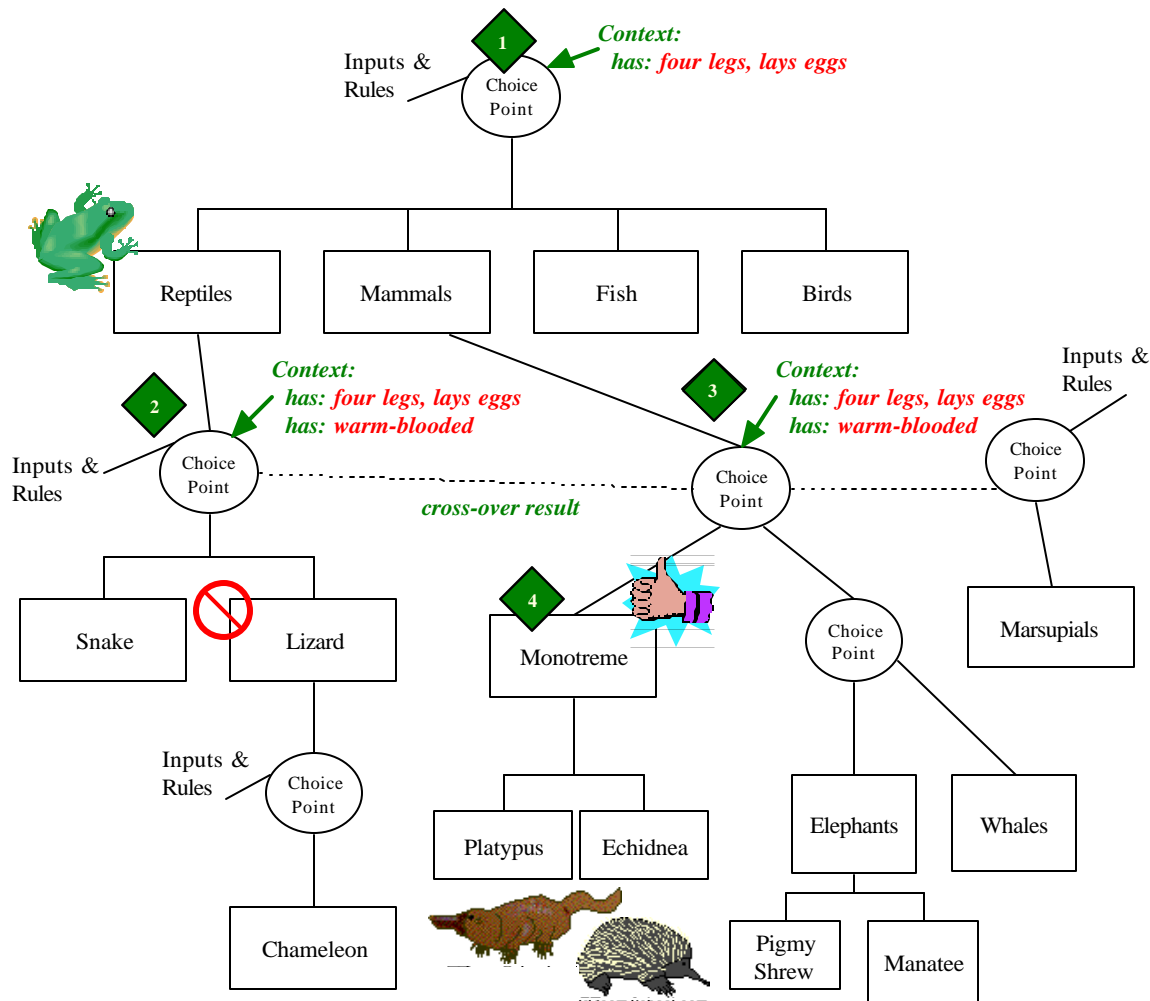
184

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

190 for an example of a contextual hierarchy) that can be traversed, and the branching that may occur
 191 across the hierarchy based on relations and associations³.

192
 193
 194

Figure 3 – contextual classification hierarchy with crossovers



195
 196
 197
 198
 199
 200
 201
 202

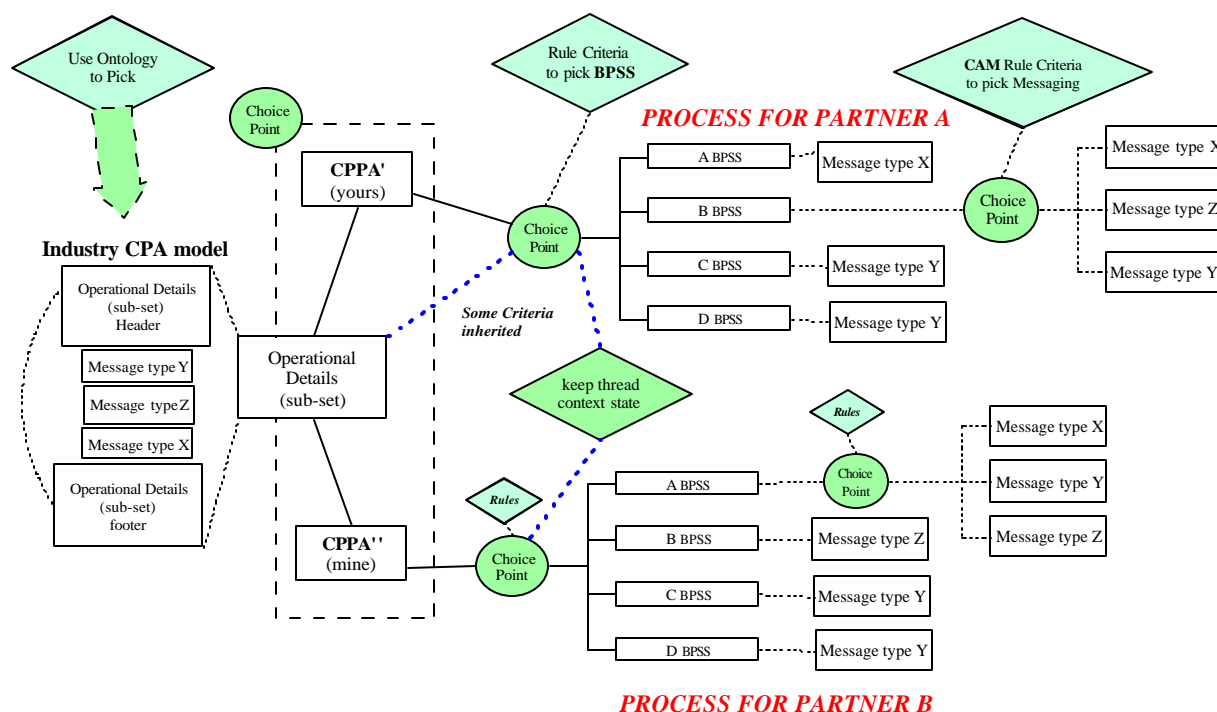
Continuing with the analysis of context types into the implementation layer from which understanding the business process is paramount. This includes process selection context and process tracking context. Below the process is the transaction context followed by the exception-handling context. At the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

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

203 This cascading of Choice Points through the business implementation layers can be seen in
 204 figure 4, where the ebXML implementation stack⁴ is used as an example. The context can be
 205 summarized as the following:

- 206
- 207 · Community of interest determination (CPPA specification / business ontology)
- 208 · Business agreement context (CPPA specification)
- 209 · Business agreement roles (CPPA specification)
- 210 · Classification of artifacts context (CPPA specification)
- 211 · Process selection context (BPSS specification)
- 212 · Process tracking context (BPSS specification)
- 213 · Transaction context (BPSS specification / CAM specification)
- 214 · Exception handling context (CAM specification)
- 215 · Decisions and rules context (CAM specification)
- 216 · Lookup tables and contextual subsets (CAM specification)

217
 218
 219 **Figure 4 – Cascading e-Business choice points within the implementation layer**
 220



221
 222
 223 Reviewing figure 4 from left to right, the initial step is to use the ontology to determine the
 224 correct community of interest and select the model for the business exchange required. The
 225 model will include details of the business process and the document exchanges (as shown with

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

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

232
233 The thread context state mechanism shown linked between the Choice Points allows both
234 partners to keep in lock step with each other's business processes as the actual exchanges occur
235 in their real-world systems (thread management is part of the Choice Point functional
236 requirements already noted earlier).

237
238 Figure 4 shows a wide variety of possible business process paths and message choices with four
239 process sequences (A,B,C,D) and three message formats (X, Y, Z). Typically business partners
240 would pick just a subset of these for their initial implementation needs.

241
242 Choice Points therefore are involved in the entire process; configuring the business partner
243 collaborations, selecting the details of the business processing, controlling the transaction content
244 messages and tracking the state of each interchange that occurs.

245
246 As previously noted the Choice Point approach lends itself to today's web service technology as
247 part of a Service Oriented Architecture (SOA). Each Choice Point can be described using XML
248 templates formatted as WSDL⁵ definitions. So in figure 3, the Choice Points denoted could easily
249 be implemented as web service driven components that provide control and selection within the
250 implementation layer.

251
252 The Choice Points could also interact with a registry of definitions so that the complete
253 behaviour can be externally configured and context driven. With such adaptability this delivers
254 agile information flows based on business context.

255

256 **5.2 Choice Point Implementation**

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

260

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

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

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

273
274 Next we consider the implementation constraints. The intention here is to provide a neutral
275 definition of the BCM Choice Point mechanisms and their XML representations that
276 implementers can then construct and integrate using popular rule engines. Since each application
277 needs will vary it is important that implementers can choose to build a tailored sub-set while
278 maintaining interoperability across Choice Points as the prime requirement.

279
280 Since Choice Points may interact themselves it is vital that the base functionality be established
281 via the use of an open XML driven service with an API (application programming interface).
282 Part of establishing this includes the ability to use a broad set of communications via WSDL
283 definitions. Other OASIS technical specifications have already successfully implemented this
284 approach, including the OASIS CAM specification. A further implementation need is that the
285 Choice Point mechanism can be used by other OASIS specifications to provide dynamic context
286 driven behaviors. Examples that have already been identified include: BPEL, BPSS, CAM,
287 CPPA, UBL, and the CIQ specifications.

288
289 In order to construct a consistent XML driven API the following components are needed:

- 290
- 291 · Rule base and consistent decision mechanisms with supporting XML syntax
- 292 · Fact base and consistent representations in XML syntax for context
- 293 · State tracking and ability to assign globally unique thread IDs
- 294 · Query and Response action formats
- 295 · Change action formats
- 296 · Event handling formats
- 297 · Security support with audit trail within the Choice Point implementation

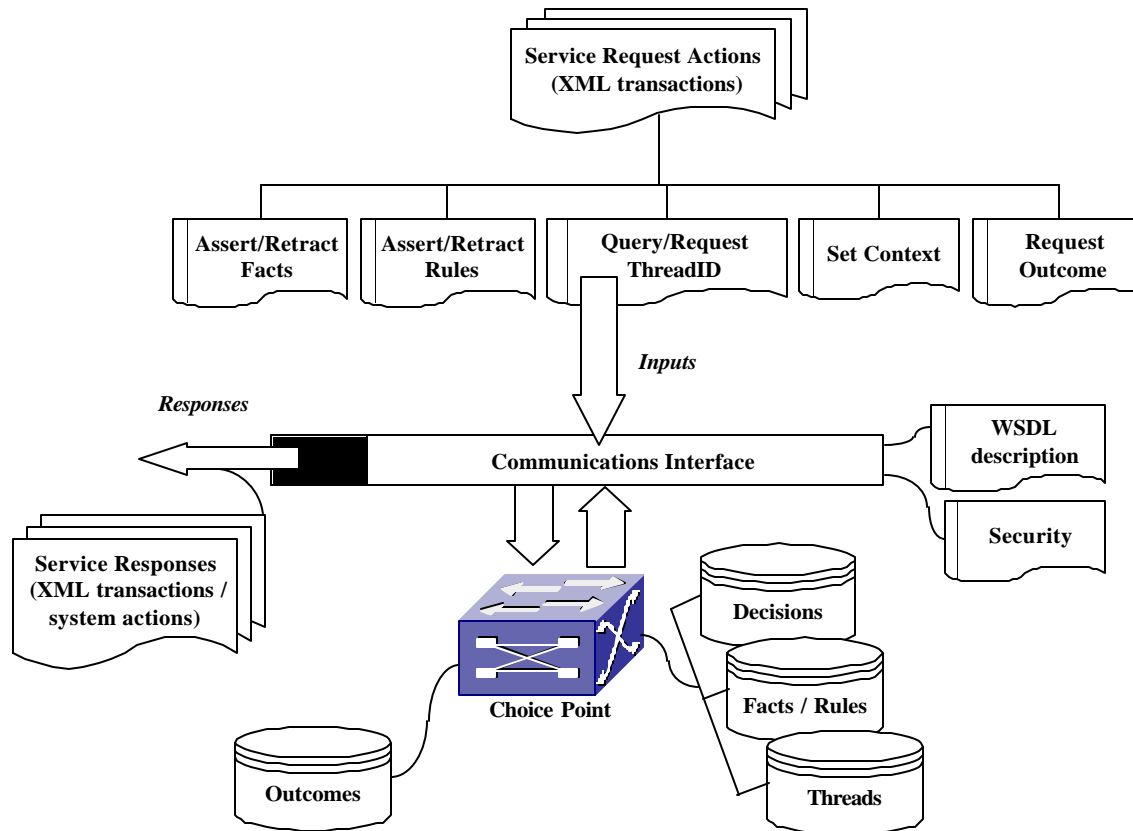
298
299 This summary is provided here, each of these items is expanded more thoroughly in the Choice
300 Point technical specification itself⁶. The primary behaviors are listed first, while those behaviors
301 likely to be optionally included in implementations are listed last.

302
303 Figure 5 depicts these components of the Choice Point implementation.

304

⁶ See BCM technical specifications for these details.

305 **Figure 5 – Choice Point rule engine implementation components**
 306



307
 308

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

314

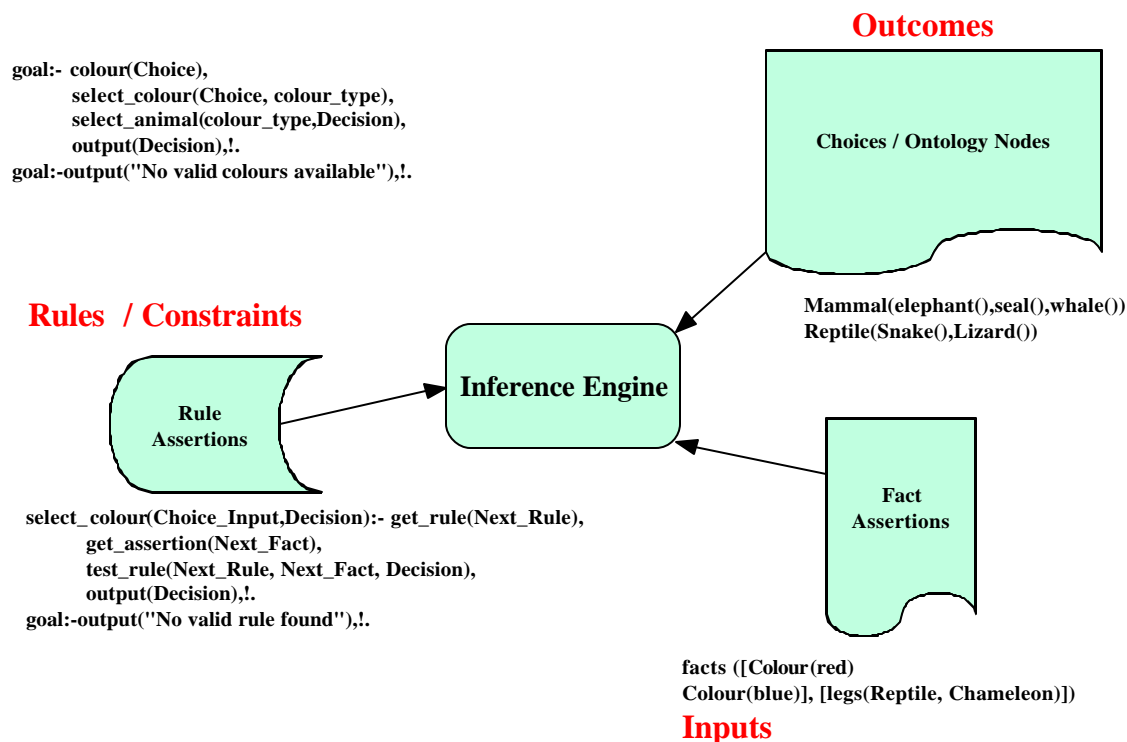
- 315 · Fact assertion / retraction
- 316 · Rule assertion / retraction
- 317 · State tracking mechanism
- 318 · Simple case rule determination (select-when-otherwise)
- 319 · Solution determination via backtracking supported
- 320 · Solution determination via forward tracking supported
- 321 · Solution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

326

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

334
 335

5.2.1.1 Figure 6 – Example of decision rules processing



336
 337

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

352 **5.3 Summary and Next Steps**

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

358
359 While decision support systems in the past have implemented such techniques they have done so
360 as closed systems. The opportunities that open rule-formats using XML together with
361 interoperable communications brings is to remove the limitations of prior architectures and
362 provide dynamic context driven implementation of enterprise systems.

363
364 This section of the BCM specifications is intended to facilitate this and form the basis for the
365 scope of action of the Linking and Switching sub-committee (SC) of the BCM technical
366 committee (TC).

367
368 It is anticipated that further liaison and outreach with other OASIS technical committees (TCs)
369 will occur to refine requirements and the implementation model, and this process has already
370 begun. Part of the deliverables for the sub-committee will include the creation of W3C WSDL
371 models for Choice Points that will help other groups to understand the interface from their own
372 specifications.

373
374 In parallel with these liaison efforts is the development of an initial Choice Points technical
375 specification details (the Pareto Principle applies!) leading to prototyping using available rule
376 engines and a demonstration using selected business scenarios.

377
378 Those interested in contributing to this work are encouraged to join the OASIS BCM TC and the
379 Linking and Switching SC, more details on this are available from the OASIS website
380 (<http://www.oasis-open.org>).