R



AMX Decisions Requirements Analysis

**Document Revisions**

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Date | Author/Department | Comments |
| 0.0 | 20 July 2010 | Paul Raby | Initial version |
| 0.1 | 21 July 2010 | Paul Raby | Cross referenced with A360; Graded Use Cases |
| 0.2 | 28 July 2010 | Paul Raby | Updated following review |
| 0.3 | 18 Aug 2010 | Paul Raby | Significant changes to requirements |
| 0.4 | 1 Sept 2010 | Paul Raby | Significant changes to requirements |
| 0.5 | 7 Sept 2010 | Paul Raby | Significant changes to architecture |
| 0.6 | 20 Sept 2010 | Paul Raby | Align with Business Events statement of direction |
| 0.7 | 4 Oct 2010 | Paul Raby | Update following review |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Copyright Notice

COPYRIGHT© 2010 TIBCO Software Inc. This document is unpublished and the foregoing notice is affixed to protect TIBCO Software Inc. in the event of inadvertent publication. All rights reserved. No part of this document may be reproduced in any form, including photocopying or transmission electronically to any computer, without prior written consent of TIBCO Software Inc. The information contained in this document is confidential and proprietary to TIBCO Software Inc. and may not be used or disclosed except as expressly authorized in writing by TIBCO Software Inc. Copyright protection includes material generated from our software programs displayed on the screen, such as icons, screen displays, and the like.

Trademarks

Technologies described herein are either covered by existing patents or patent applications are in progress. All brand and product names are trademarks or registered trademarks of their respective holders and are hereby acknowledged.

Confidentiality

The information in this document is subject to change without notice. This document contains information that is confidential and proprietary to TIBCO Software Inc. and may not be copied, published, or disclosed to others, or used for any purposes other than review, without written authorization of an officer of TIBCO Software Inc. Submission of this document does not represent a commitment to implement any portion of this specification in the products of the submitters.

Content Warranty

The information in this document is subject to change without notice. THIS DOCUMENT IS PROVIDED "AS IS" AND TIBCO MAKES NO WARRANTY, EXPRESS, IMPLIED, OR STATUTORY, INCLUDING BUT NOT LIMITED TO ALL WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. TIBCO Software Inc. shall not be liable for errors contained herein or for incidental or consequential damages in connection with the furnishing, performance or use of this material.

For more information, please contact:

TIBCO Software Inc.  
3303 Hillview Avenue  
Palo Alto, CA 94304  
USA

Contents

[1 Scope 7](#_Toc274041572)

[1.1 Assertions 7](#_Toc274041573)

[1.2 Phasing 7](#_Toc274041574)

[2 Generic In Use Story 9](#_Toc274041575)

[3 Phase Zero: How To Guide Using Mediation 10](#_Toc274041576)

[3.1 Summary 10](#_Toc274041577)

[3.2 Requirements 10](#_Toc274041578)

[3.3 Non-Requirements 10](#_Toc274041579)

[3.4 Restrictions 10](#_Toc274041580)

[3.5 Typical Use Case 10](#_Toc274041581)

[4 Phase One: How To Guide Without Using Mediation 12](#_Toc274041582)

[4.1 Summary 12](#_Toc274041583)

[4.2 Requirements 12](#_Toc274041584)

[4.3 Non-Requirements 12](#_Toc274041585)

[4.4 Restrictions 12](#_Toc274041586)

[4.5 Typical Use Case 12](#_Toc274041587)

[5 Phase Two: (Variant One) AMX Decisions with Simple Decision Flow (AMX BPM) 14](#_Toc274041588)

[5.1 Summary 14](#_Toc274041589)

[5.2 Requirements 14](#_Toc274041590)

[5.3 Non-Requirements 14](#_Toc274041591)

[5.4 Typical Use Case 14](#_Toc274041592)

[6 Phase Two: (Variant Two) AMX Decisions with Simple Decision Flow (non AMX BPM) 16](#_Toc274041593)

[6.1 Summary 16](#_Toc274041594)

[6.2 Requirements 16](#_Toc274041595)

[6.3 Non-Requirements 16](#_Toc274041596)

[6.4 Typical Use Case 16](#_Toc274041597)

[7 Phase Three: AMX Decisions with Enhanced Decision Flow 18](#_Toc274041598)

[7.1 Summary 18](#_Toc274041599)

[7.2 Requirements 18](#_Toc274041600)

[7.3 Non-Requirements 18](#_Toc274041601)

[7.4 Typical Use Case 18](#_Toc274041602)

[8 Phase Four: AMX Decisions with Post-Deployment Rule Modification 20](#_Toc274041603)

[8.1 Summary 20](#_Toc274041604)

[8.2 Requirements 20](#_Toc274041605)

[8.3 Non-Requirements 20](#_Toc274041606)

[8.4 Typical Use Case 20](#_Toc274041607)

[9 Architecture 22](#_Toc274041608)

[9.1 Overview 22](#_Toc274041609)

[9.2 Phase Two: AMX Decisions With Simple Decision Flow 22](#_Toc274041610)

[9.2.1 Generic AMX Decisions 22](#_Toc274041611)

[9.2.2 AMX BPM Specific 23](#_Toc274041612)

[9.2.3 Generic Business Studio Add-Ins 24](#_Toc274041613)

[9.3 Phase Three: AMX Decisions With Enhanced Decision FLow 24](#_Toc274041614)

[9.3.1 Generic AMX Decisions 24](#_Toc274041615)

[9.3.2 AMX BPM Specific 25](#_Toc274041616)

[9.3.3 Generic Business Studio Add-Ins 25](#_Toc274041617)

[9.4 Phase Four: AMX Decisions With Post-Deployment Rule Modification 25](#_Toc274041618)

[9.4.1 Generic AMX Decisions 25](#_Toc274041619)

[9.4.2 AMX BPM Specific 26](#_Toc274041620)

[9.4.3 Generic Business Studio Add-Ins 26](#_Toc274041621)

[9.4.4 Administration 26](#_Toc274041622)

[10 Mapping of Architectural Components to Use Cases 27](#_Toc274041623)

[10.1 Phase Two: (Variant One) AMX Decisions with Simple Decision Flow (AMX BPM) 27](#_Toc274041624)

[10.1.1 Deliverables 27](#_Toc274041625)

[10.1.2 Deployment 27](#_Toc274041626)

[10.1.3 Installation 27](#_Toc274041627)

[10.2 Phase Two: (Variant Two) AMX Decisions with Simple Decision Flow (non AMX BPM) 27](#_Toc274041628)

[10.2.1 Deliverables 27](#_Toc274041629)

[10.2.2 Deployment 27](#_Toc274041630)

[10.2.3 Installation 27](#_Toc274041631)

[10.3 Phase Three: AMX Decisions with Enhanced Decision Flow 27](#_Toc274041632)

[10.3.1 Deliverables 27](#_Toc274041633)

[10.3.2 Deployment 28](#_Toc274041634)

[10.3.3 Installation 28](#_Toc274041635)

[10.4 Phase Four: AMX Decisions with Post-Deployment Rule Modification 28](#_Toc274041636)

[10.4.1 Deliverables 28](#_Toc274041637)

[10.4.2 Deployment 28](#_Toc274041638)

[10.4.3 Installation 28](#_Toc274041639)

[11 Dependencies and Requirements 29](#_Toc274041640)

[11.1 Phase Zero: A How to Guide Using Mediation 29](#_Toc274041641)

[11.2 Phase One: A How to Guide Without Using Mediation 29](#_Toc274041642)

[11.2.1 Business Events 29](#_Toc274041643)

[11.2.2 AMX Platform 29](#_Toc274041644)

[11.2.3 AMX BPM 29](#_Toc274041645)

[11.2.4 BPM Business Studio 29](#_Toc274041646)

[11.2.5 BOM MetaData 29](#_Toc274041647)

[11.2.6 Generic Business Studio 29](#_Toc274041648)

[11.2.7 Administrator GUI 29](#_Toc274041649)

[11.3 Phase Two: AMX Decisions with Simple Decision Flow 29](#_Toc274041650)

[11.3.1 Business Events 29](#_Toc274041651)

[11.3.2 AMX Platform 29](#_Toc274041652)

[11.3.3 AMX BPM 30](#_Toc274041653)

[11.3.4 BPM Business Studio 30](#_Toc274041654)

[11.3.5 BOM MetaData 30](#_Toc274041655)

[11.3.6 Generic Business Studio 30](#_Toc274041656)

[11.3.7 Administrator GUI 30](#_Toc274041657)

[11.4 Phase Three: AMX Decisions with Enhanced Decision Flow 30](#_Toc274041658)

[11.4.1 Business Events 30](#_Toc274041659)

[11.4.2 AMX Platform 30](#_Toc274041660)

[11.4.3 AMX BPM 30](#_Toc274041661)

[11.4.4 BPM Business Studio 30](#_Toc274041662)

[11.4.5 BOM MetaData 31](#_Toc274041663)

[11.4.6 Generic Business Studio 31](#_Toc274041664)

[11.4.7 Administrator GUI 31](#_Toc274041665)

[11.5 Phase Four: AMX Decisions with Post-Deployment Rule Modification 31](#_Toc274041666)

[11.5.1 Business Events 31](#_Toc274041667)

[11.5.2 AMX Platform 31](#_Toc274041668)

[11.5.3 AMX BPM 31](#_Toc274041669)

[11.5.4 BPM Business Studio 31](#_Toc274041670)

[11.5.5 BOM MetaData 32](#_Toc274041671)

[11.5.6 Generic Business Studio 32](#_Toc274041672)

[11.5.7 Administrator GUI 32](#_Toc274041673)

[12 Accept 360 Requirements Reference 33](#_Toc274041674)

[13 Appendix A: Glossary 34](#_Toc274041675)

# Scope

This document is produced by the engineering team to try and reconcile the requested requirements with what is realistic and achievable based on a phased delivery.

There is a glossary of terminology contained in Appendix A: Glossary.

## Assertions

1. The intention is that AMX Decisions is a top level TIBCO product independent of AMX BPM with no dependency on AMX BPM. In other words AMX Decisions should be capable of being used within the AMX context without AMX BPM being installed. It is *not* AMX BPM Decisions.
2. It will be possible to invoke a Decision Service from within AMX BPM but also from any other AMX product, e.g. AMX Service Grid, or via web services from a non AMX product, e.g. iProcess Engine. AMX BPM in this scenario will likely not be deployed.
3. For the purpose of AMX Decisions, BusinessEvents will be treated as an internal OEM. This directly implies that there is a hard link to a specific (possibly bespoke) version and that there is no requirement to link to or maintain on-going parity with the latest BusinessEvents release.
4. AMX Decisions will only expose the Decision Table, Concept Model and Domain Model functionality of TIBCO BusinessEvents. All other functionality, including but not restricted to rules, events, channels, etc., will be either restricted or not included in the product.
5. The invocation of the Decision Tables will be stateless.
6. The BusinessEvents Concept Model will be generated from the Business Object Model (BOM). The BOM is the single source of truth and changes to the BOM will be reflected in the Concept Model. The generated Concept Model will be read-only with changes being initiated via changes to the BOM. The BOM will be able to be reused across BDS, CDOS, etc.
7. Domain Models will be expressed as enumerations on the Business Object Model. There will be no new requirements for the BOM however this will result in a Domain Model that has a subset of functionality compared to that in a BusinessEvents domain model. (Note: Domain models are optional).
8. There will be little or no dependency on changes to the AMX BPM Process Engine in order to deliver AMX Decisions functionality.
9. Versioning of Decision Services must be supported.

## Phasing

The requirements will be realised in five phases, these being:

* Phase Zero: A How to Guide Using Mediation
* Phase One: A How to Guide Without Using Mediation
* Phase Two: AMX Decisions with Simple Decision Flow
* Phase Three: AMX Decisions with Enhanced Decision Flow
* Phase Four: AMX Decisions with Post-Deployment Rule Modification

Note that phases two and three are mutually exclusive and much of the effort required to implement phase two will be negated once phase three is implemented. Phase four may be combined into phase three depending on the degree of overlap in the AMX Decisions and BusinessEvents product roadmap and delivery schedule.

Further note that a lesser integration (Phases Zero and One) provides a more generic integration between BPM and BE and permits the full exposure of BE functionality which may be a requirement for certain customer engagements. It is not however AMX Decisions.

# Generic In Use Story

This remains as per the original specification in requirement [#82883](https://tibco.acceptondemand.com/gotoEntity?entityType=ENTITY_WITH_OWNER&entityId=82883). Note however that this scenario does involve AMX BPM whereas it is perfectly acceptable for AMX Decisions to be called from AMX Service Grid, for example, without AMX BPM.

|  |
| --- |
| XYZ Healthcare Insurance is implementing a New Policy Administration system using TIBCO's ActiveMatrix BPM and ActiveMatrix Decisions solutions.  Applicants can fill out applications for new policies on-line where they will be given immediate feedback as to whether they have been accepted.  As an early part of the "New Policy" process there is a check that validates whether the applicant meets their basic criteria for the policy for which they are applying.  This check looks at a number of factors relating to the applicant, including   * Age * Sex * Smoking * Previous Conditions * Previous Claims   This returns a risk factor that is used to control down-stream processing. XYZ have decided to implement the risk scoring check as a service as they wish to re-use it from multiple locations and they wish to lifecycle the rules independent of the business processes.  The service returns one of four risk factors, 0-3 where 0 means the applicant is not qualified for the policy. The applicant is told then-and-there that they are not qualified for the requested product. Applications with risk factor 3 are told then-and-there that they have been accepted. For applications for risk factors 1 and 2 the applicant is informed that their application requires review before acceptance whereby the process continues off-line from the applicant and the application is routed to specialist under-writing teams dependent upon the risk factor and applicant data. |

# Phase Zero: How To Guide Using Mediation

## Summary

This is a fully manual integration between BPM and BE with no significant auto-generation of artefacts.

## Requirements

AMX Mediation component to integrate the differing web service standard capabilities between BPM and BE.

## Non-Requirements

No changes necessary to Business Studio, AMX BPM, or Business Events.

## Restrictions

XPD-950 currently limits the scope of construction of data objects.

## Typical Use Case

This is a new specification variant of requirement [#82883](https://tibco.acceptondemand.com/gotoEntity?entityType=ENTITY_WITH_OWNER&entityId=82883).

|  |
| --- |
| The SME (Subject Matter Expert), the BE developer and the BPM developer are working using a JAD-style of development within two DEVELOPMENT environments. This means they are using BPM Business Studio and BusinessEvents Studio to deploy artefacts to independent infrastructures for processes and rules.  Primary Actors: SME, BPM Developer, BE Developer Secondary Actors: BPM Business Studio, BusinessEvents Studio, AMX-BPM and BE  **BPM Business Studio**   1. BPM Developer creates a BPM project. 2. BPM Developer and SME agree a Business Object Model. This is created using the BPM Business Studio BOM. 3. BPM Developer adds a Web Service activity to the process model. 4. BPM Developer generates a WSDL based on the BOM and web service activity.    * Service parameters are auto-mapped. 5. BPM Developer concretises the WSDL using RPC/Literal binding.   **BusinessEvents Studio**   1. BE Developer creates BusinessEvents project. 2. BE Developer and SME creates Concepts (manually). 3. BE Developer creates an XML Schema for use in the Web Service interface. 4. BE Developer creates Shared Resources, Channels and Events. 5. BE Developer creates Rule Functions and Rules to interface with the Decision Service. 6. SME creates Decision Service definition (hierarchy of Decision Tables) using BE Decision Manager UI and Concepts defined earlier. 7. SME Tests and validates Decision Service standalone using existing tooling. 8. BE Developer deploys Decision Service to development BE environment using BE tooling (BE runtime is not part of ActiveMatrix runtime). 9. BE Developer exports WSDL from BusinessEvents project.   **Mediation (in BPM Business Studio)**   1. BPM Developer creates SOA project. 2. BPM Developer imports BPM and BE WSDL’s. 3. BPM Developer creates a mediation flow to join the BPM and BE web services.    * Parameters are manually mapped using the mapper. 4. BPM Developer creates a composite and mediation SOA component. 5. BPM Developer creates a concrete binding for the BE web service interface on the component. 6. BPM Developer deploys the component to AMX, binding of BE service is validated on deployment.   **BPM Business Studio**   1. (Optional) BPM Developer uses Tester/Debugger to test decision service invocation (see dedicated use case). 2. BPM Developer deploys Business Process; binding of mediation service is validated on deployment.   **Runtime**   1. BPM Developer executes the process. 2. Starts New Policy business service. 3. Fills in applicant details on first form. 4. Decision Service is invoked. 5. Results are displayed to end user. |

# Phase One: How To Guide Without Using Mediation

**Note: it is not possible to test this use case for accuracy until top level element support is available. Modifications to AMX BPM are pending in order to support this use case.**

## Summary

This is a fully manual integration between BPM and BE with *significant* auto-generation of artefacts.

## Requirements

Top level element support in AMX BPM including Business Studio.

## Non-Requirements

No changes necessary to Business Events.

## Restrictions

XPD-950 currently limits the scope of construction of data objects.

## Typical Use Case

This is a new specification variant of requirement [#82883](https://tibco.acceptondemand.com/gotoEntity?entityType=ENTITY_WITH_OWNER&entityId=82883).

|  |
| --- |
| The SME (Subject Matter Expert), the BE developer and the BPM developer are working using a JAD-style of development within two DEVELOPMENT environments. This means they are using BPM Business Studio and BusinessEvents Studio to deploy artefacts to independent infrastructures for processes and rules.  Primary Actors: SME, BPM Developer, BE Developer Secondary Actors: BPM Business Studio, BusinessEvents Studio, AMX-BPM and BE  **BPM Business Studio**   1. BPM Developer creates a BPM project. 2. BPM Developer and SME agree a Business Object Model. This is created using the BPM Business Studio BOM. 3. BPM Developer adds a Web Service activity to the process model. 4. BPM Developer generates a WSDL based on the BOM and web service activity.    * Service parameters are auto-mapped. 5. BPM Developer concretises the WSDL using Document/Literal binding.   **BusinessEvents Studio**   1. BE Developer creates BusinessEvents project. 2. BE Developer imports the BPM WSDL and auto-generates BE artefacts. 3. SME creates Decision Service definition (hierarchy of Decision Tables) using BE Decision Manager UI and Concepts defined earlier. 4. SME Tests and validates Decision Service standalone using existing tooling. 5. BE Developer deploys Decision Service to development BE environment using BE tooling (BE runtime is not part of ActiveMatrix runtime).   **BPM Business Studio**   1. (Optional) BPM Developer uses Tester/Debugger to test decision service invocation (see dedicated use case). 2. BPM Developer deploys Business Process; binding of mediation service is validated on deployment.   **Runtime**   1. BPM Developer executes the process. 2. Starts New Policy business service. 3. Fills in applicant details on first form. 4. Decision Service is invoked. 5. Results are displayed to end user. |

# Phase Two: (Variant One) AMX Decisions with Simple Decision Flow (AMX BPM)

## Summary

This is a realisation of the full use case for AMX Decisions within a BPM context. The Decision Flow is restricted to purely decision tables. It is a direct runtime integration into the BE decision table functionality with no additional overhead of events, channels, etc.

## Requirements

1. BOM annotations for BE Concept meta-data in Business Studio.
2. BE concept generator for Business Studio (one-way, BPM BOM to BE Concept with the BOM being the single source of truth).
3. Add-in Studio Feature – ability to install selected elements of BE Studio and (BPM) Business Studio side-by-side to create a single consolidated studio environment.
4. BE API for Studio to generate BE concepts and domain models programmatically. (Note: domain model support is optional)
5. BE runtime accessible via Java API to manipulate concepts and directly invoke a decision service without the need for shared resources, channels, events, rules, etc.
6. Decision Flow modeller (similar to Page Flow) to combine one or more decision tables into a decision service.
7. Decision Service Activity for BPM Business Studio.
8. Decision Flow runtime capability built as the AMX Decisions component.

## Non-Requirements

No requirements for web service aspects of BusinessEvents.

## Typical Use Case

This is a modification to the original specification in requirement [#82883](https://tibco.acceptondemand.com/gotoEntity?entityType=ENTITY_WITH_OWNER&entityId=82883).

Note that this scenario does involve AMX BPM whereas it is perfectly acceptable for AMX Decisions to be called from AMX Service Grid, for example, without AMX BPM.

|  |
| --- |
| The SME (Subject Matter Expert) and the BPM developer are working using a JAD-style of development within a DEVELOPMENT environment. This means they are using Business Studio to deploy artefacts to a shared infrastructure for processes and rules.  Primary Actors: SME, BPM Developer Secondary Actors: Business Studio with AMX Decisions Add-In, AMX-BPM and BusinessEvents  **Business Studio**   1. BPM Developer creates a BPM project. 2. (Optional but normal) SME creates Decision Service project. 3. BPM Developer and SME agree a Business Object Model. This is created using the Business Studio BOM with BE meta-data annotations added. 4. BPM Developer defines a Decision Service activity in the process model. 5. BPM Developer adds BOM data references from the process model to the Decision Service activity interface definition. The interface for the Decision Service specifies the process data to be made available, and the process data in turn references BOM types as externally defined data types. 6. BPM Developer generates a Decision Flow based on the data specified in the interface. The Decision Flow will contain a single ‘blank’ Decision Table activity. Typically this will be achieved by context-selecting the Decision Service activity and selecting ‘Generate Decision Flow’. 7. Business Studio auto-generates Concepts from the BOM data reference and makes the Concepts available in the Decision Flow. This involves converting BOM types to Concept types based on the BOM type definitions. 8. SME populates the Decision Flow with one or more Decision Tables. Calls to Java Script activities, gateways, or other activity types, e.g. web services, and the use of events, is restricted from use. 9. SME populates Decision Tables using BE Decision Manager UI and the read-only Concepts auto-generated from BOM. 10. SME Tests and validates Decision Tables standalone using existing tooling. 11. SME deploys Decision Service. 12. (Optional) BPM Developer uses Tester/Debugger to test decision service invocation (see dedicated use case). 13. BPM Developer deploys Business Process and Decision Service.   **Runtime**   1. BPM Developer executes the process. 2. Starts New Policy business service. 3. Fills in applicant details on first form. 4. Decision Service is invoked. 5. Results are displayed to end user. |

# Phase Two: (Variant Two) AMX Decisions with Simple Decision Flow (non AMX BPM)

## Summary

This is a realisation of the full use case for AMX Decisions outside of a BPM context. The Decision Flow is restricted to purely decision tables. It is a direct runtime integration into the BE decision table functionality with no additional overhead of events, channels, etc.

## Requirements

1. BOM annotations for BE Concept meta-data in Business Studio.
2. BE concept generator for Business Studio (one-way, BPM BOM to BE Concept with the BOM being the single source of truth).
3. Add-in Studio Feature – ability to install selected elements of BE Studio and (AMX) Business Studio side-by-side to create a single consolidated studio environment.
4. BE API for Studio to generate BE concepts and domain models programmatically. (Note: domain model support is optional)
5. BE runtime accessible via Java API to manipulate concepts and directly invoke a decision service without the need for shared resources, channels, events, rules, etc.
6. Decision Flow modeller (similar to Page Flow) to combine one or more decision tables into a decision service.
7. Decision Flow runtime capability built as the AMX Decisions component.

## Non-Requirements

No requirements for web service aspects of BusinessEvents.

## Typical Use Case

This is a modification to the original specification in requirement [#82883](https://tibco.acceptondemand.com/gotoEntity?entityType=ENTITY_WITH_OWNER&entityId=82883).

|  |
| --- |
| The SME (Subject Matter Expert) and the AMX developer are working using a JAD-style of development within a DEVELOPMENT environment. This means they are using Business Studio to deploy artefacts to a shared infrastructure for processes and rules.  Primary Actors: SME, AMX Developer Secondary Actors: Business Studio with AMX Decisions Add-In, AMX and BusinessEvents  **Business Studio**   1. AMX Developer creates an AMX project. 2. (Optional but normal) SME creates Decision Service project. 3. AMX Developer and SME agree a Business Object Model. This is created using the Business Studio BOM with BE meta-data annotations added. 4. AMX Developer generates a Decision Flow specifying BOM data references as required. The Decision Flow will contain a single ‘blank’ Decision Table activity. 5. Business Studio auto-generates Concepts from the BOM data references and makes the Concepts available in the Decision Flow. The generation is synchronised with the addition or removal of BOM data references. 6. SME populates the Decision Flow with one or more Decision Tables. Calls to Java Script activities, gateways, or other activity types, e.g. web services, and the use of events, is restricted from use. 7. SME populates Decision Tables using BE Decision Manager UI and the read-only Concepts auto-generated from BOM. 8. SME Tests and validates Decision Tables standalone using existing tooling. 9. SME deploys Decision Service. 10. AMX Developer binds to the generic AMX Decisions web service interface and maps parameters as necessary. 11. AMX Developer deploys the Decision Service.   **Runtime**   1. AMX Developer executes the containing service. 2. Decision Service is invoked. 3. Results are returned to the containing service. |

# Phase Three: AMX Decisions with Enhanced Decision Flow

## Summary

This is a realisation of the full use case for AMX Decisions within a BPM context. It can equally apply to the generic AMX context outside of AMX BPM. The Decision Flow is enhanced to support decision tables, JavaScript[\*](#JavaScript1) activities and gateways using BPMN modelling. It remains a direct runtime integration of the BE decision table functionality with no additional overhead of events, channels, etc.

## Requirements

1. BOM annotations for BE Concept meta-data in Business Studio.
2. BE concept generator for Business Studio (one-way, BPM BOM to BE Concept with the BOM being the single source of truth).
3. Add-in Studio Feature – ability to install selected elements of BE Studio and (BPM) Business Studio side-by-side to create a single consolidated studio environment.
4. BE API for Studio to generate BE concepts and domain models programmatically. (Note: domain model support is optional)
5. BE runtime accessible via Java API to manipulate concepts and directly invoke a decision service without the need for shared resources, channels, events, rules, etc.
6. Decision Flow modeller (similar to Page Flow) to model one or more decision tables as a decision service in combination with JavaScript[\*](#JavaScript1) activities and gateways.
7. Decision Service Activity for BPM Business Studio.
8. Decision Flow runtime capability built as the AMX Decisions component.

## Non-Requirements

No requirements for web service aspects of BusinessEvents.

## Typical Use Case

This is a modification to the original specification in requirement [#82883](https://tibco.acceptondemand.com/gotoEntity?entityType=ENTITY_WITH_OWNER&entityId=82883).

Note that this scenario does involve AMX BPM whereas it is perfectly acceptable for AMX Decisions to be called from AMX Service Grid, for example, without AMX BPM.

|  |
| --- |
| The SME (Subject Matter Expert) and the BPM developer are working using a JAD-style of development within a DEVELOPMENT environment. This means they are using Business Studio to deploy artefacts to a shared infrastructure for processes and rules.  Primary Actors: SME, BPM Developer Secondary Actors: Business Studio with AMX Decisions Add-In, AMX-BPM and BusinessEvents  This is an extension use case of Phase Two Variant One.  **Business Studio**   1. BPM Developer creates a BPM project. 2. (Optional but normal) SME creates Decision Service project. 3. BPM Developer and SME agree a Business Object Model. This is created using the Business Studio BOM with BE meta-data annotations added. 4. BPM Developer defines a Decision Service activity in the process model. 5. BPM Developer adds BOM data references from the process model to the Decision Service activity interface definition. The interface for the Decision Service specifies the process data to be made available, and the process data in turn references BOM types as externally defined data types. 6. BPM Developer generates a Decision Flow based on the data specified in the interface. The Decision Flow will contain a single ‘blank’ Decision Table activity. Typically this will be achieved by context-selecting the Decision Service activity and selecting ‘Generate Decision Flow’. 7. Business Studio auto-generates Concepts from the BOM data reference and makes the Concepts available in the Decision Flow. This involves converting BOM types to Concept types based on the BOM type definitions. 8. SME populates the Decision Flow with one or more Decision Tables, Java Script activities[\*](#JavaScript1), or gateways. Other activity types, e.g. web services, and the use of events, remains restricted from use. 9. SME populates Decision Tables using BE Decision Manager UI and the read-only Concepts auto-generated from BOM. 10. SME Tests and validates Decision Tables standalone using existing tooling. 11. SME deploys Decision Service. 12. (Optional) BPM Developer uses Tester/Debugger to test decision service invocation (see dedicated use case). 13. BPM Developer deploys Business Process and Decision Service.   **Runtime**   1. BPM Developer executes the process. 2. Starts New Policy business service. 3. Fills in applicant details on first form. 4. Decision Service is invoked. 5. Results are displayed to end user. |

*\* The use of JavaScript is dependent on this being supported by BusinessEvents in the context of a Decision Graph. It would enable the setting of local data variables prior to their reference by a Decision Table. Such a capability will need to be evaluated with the BusinessEvents team.*

# Phase Four: AMX Decisions with Post-Deployment Rule Modification

## Summary

This is a realisation of the full use case for AMX Decisions within a BPM context. It can equally apply to the generic AMX context outside of AMX BPM. The ability to modify the contents of a decision table post-deployment is introduced. It remains a direct runtime integration of the BE decision table functionality with no additional overhead of events, channels, etc.

## Requirements

1. BOM annotations for BE Concept meta-data in Business Studio.
2. BE concept generator for Business Studio (one-way, BPM BOM to BE Concept with the BOM being the single source of truth).
3. Add-in Studio Feature – ability to install selected elements of BE Studio and (BPM) Business Studio side-by-side to create a single consolidated studio environment.
4. BE API for Studio to generate BE concepts and domain models programmatically. (Note: domain model support is optional)
5. BE runtime accessible via Java API to manipulate concepts and directly invoke a decision service without the need for shared resources, channels, events, rules, etc.
6. Decision Flow modeller (similar to Page Flow) to model one or more decision tables as a decision service in combination with JavaScript[\*](#JavaScript1) activities and gateways.
7. Decision Service Activity for BPM Business Studio.
8. Decision Flow runtime capability built as the AMX Decisions component.
9. Administration interface (possibly BE Decision Manager) to author and deploy changes to decision tables post-deployment.
10. Extension to AMX Decisions component to support changes to decision tables post-deployment.

## Non-Requirements

No requirements for web service aspects of BusinessEvents.

## Typical Use Case

This is a modification to the original specification in requirement [#82883](https://tibco.acceptondemand.com/gotoEntity?entityType=ENTITY_WITH_OWNER&entityId=82883).

Note that this scenario does involve AMX BPM whereas it is perfectly acceptable for AMX Decisions to be called from AMX Service Grid, for example, without AMX BPM.

|  |
| --- |
| The SME (Subject Matter Expert) and the BPM developer are working using a JAD-style of development within a DEVELOPMENT environment. This means they are using Business Studio to deploy artefacts to a shared infrastructure for processes and rules.  Primary Actors: SME, BPM Developer Secondary Actors: Business Studio with AMX Decisions Add-In, AMX-BPM and BusinessEvents  This is an extension use case of Phase Three.  **Business Studio**   1. BPM Developer deploys Business Process and Decision Service as previous use case.   **Runtime**   1. BPM Developer executes the process as previous use case. 2. SME modifies decision table via the administration interface. 3. BPM Developer executes the process as previous use case, which uses the modified decision service. |

# Architecture

This section relates purely to AMX Decisions (Phases Two, Three and Four).

## Overview

There are four distinct architectural divisions. These are the:

* Generic AMX Decisions product;
* AMX BPM specific features, including BPM Business Studio features;
* Generic Business Studio features;
* Administration features.

The architecture changes as each subsequent phase is implemented. As such the architecture is described progressively with respect to each phase.

## Phase Two: AMX Decisions With Simple Decision Flow

### Generic AMX Decisions

*Note: this section describes the generic runtime features. There must also be a complimentary design time feature – refer to subsequent sections for more detail.*

The generic AMX features are based on a standard SCA component based on a Spring implementation type (Spring IT). It is composed as shown below.

Minimal Business Events Engine

ADEC Wrapper

DEP

DEP

DEP

**AMX Decisions SCA Component**

Evaluate

The component contains:

* Minimal BusinessEvents Engine – enables direct processing of a decision service defined as a rule function and based on concept data models.
* ADEC Wrapper – the AMX Decisions wrapper controls and correlates between the BusinessEvents engine, deployed artefacts and the external SCA interface.
* DEP – the deployed XML bundle of rule function, decision tables, associated concepts, data mapping (if applicable) and BDS schema.

The component also acts as a deployment target.

The external interface is based on other similar interfaces as used in Business Resource Management. The Evaluate interface is static and is used for invocation based on BOM objects described using the *XMLAny* type. It is ideally suitable for invoking via AMX BPM. It can also be invoked from a more generic client, such as AMX Service Grid or iProcess, that construct the SOAP payload directly and solely from the WSDL definition of the interface. This however this would best be achieved using mediation to present a strongly typed bespoke interface that does not use *XMLAny*.

The lifecycle of the component is that at node start-up the BusinessEvents engine is initialised and deployed artefacts are instantiated such that there is minimal performance impact when executing a specific decision table. As additional deployments occur these are also instantiated with the BusinessEvents engine.

The ADEC wrapper hydrates BusinessEvents concepts based on a combination of the data specified in the XMLAny input and deployed mapping data and invokes the deployed Rule Function in the BusinessEvents engine. This in turn evaluates each referenced Decision Table. The resulting concepts are dehydrated and serialised back via the external interface as XML.

### AMX BPM Specific

There is a runtime and a design time aspect to the BPM specific features.

At design time there is a custom Decisions activity type for BPM BusinessStudio. There is also a facility to auto-generate BusinessEvents concepts from BOM objects in a one-way read-only transformation when creating a Decision Flow.

The decisions activity, when using BOM data in the process or page flow, will automatically map the BOM data to Concept data as input and output mappings. If using simple data types the mapping must be completed manually, similarly to when using a Web Service activity type.

One or more Decision Table activities can be inserted into the Decision Flow. No other construct is permitted in this phase.

When deployed the Decision Flow is converted by Business Studio to a BE rule function that calls each Decision Table in the flow in turn. When deploying from Business Studio the BusinessEvents concepts, decision tables, input/output mapping for the decisions activity and (optionally) the BOM generated BDS schema (if using complex data) are deployed as a package to the Decisions SCA Component deployment target.

At runtime the decisions activity is handled by the process engine in the same way as User Task is currently handled. A Task Extension Point is created that is wired to the generic AMX Decisions SCA component as shown below.

**AMX Decisions SCA Component**

Evaluate

Process Template

ADEC TEP

**Process Engine Component**

At runtime the Decisions TEP retrieves the data values from the process engine and then serialises the data as XML to the AMX Decisions SCA component via the Evaluate interface. This serialisation may use either BDS (if using complex data) or name/value pairs if using simple data.

Following execution of the decision service the TEP updates the process engine data based on the results returned by the component, using BDS if necessary.

### Generic Business Studio Add-Ins

At design time there is a facility to auto-generate BusinessEvents concepts from BOM objects in a one-way read-only transformation when creating a Decision Flow.

At runtime the web service is invoked via the generic AMX Decisions SCA component. This could be wrapped in a bespoke service via the use of mediation if desired.

## Phase Three: AMX Decisions With Enhanced Decision FLow

### Generic AMX Decisions

*Note: this section describes the generic runtime features. There must also be a complimentary design time feature – refer to subsequent sections for more detail.*

In this phase the AMX Decisions SCA component is retained but operates differently.

The deployed artefacts contain the deployed XML bundle of decision tables, associated concepts, data mapping (if applicable) and BDS schema as previously. However the rule function is replaced by a Decision Graph definition, the BusinessEvents BPMN model of a Decisions Flow.

As a consequence the ADEC wrapper and interface into the BusinessEvents engine is modified to use the Decision Graph rather than the rule function as the definition of the flow to be executed.

The component continues to act as a deployment target. The external interface and lifecycle behaviour is as per the previous phase.

### AMX BPM Specific

The Decisions Activity and concept generation capability in Business Studio continues in this phase. However the Decisions Flow is now modelled using BPMN constructs containing one or more Decision Table activities, JavaScript[\*](#JavaScript1) activities and gateways. No other construct is permitted.

When deployed the XPDL for the Decision Flow is converted by Business Studio to the equivalent XPDL for a BusinessEvents Decision Flow. This in turn is converted by BusinessEvents to the BusinessEvents Decision Graph definition. This is deployed along with the other artefacts as in the previous phase to the Decisions SCA Component deployment target.

With respect to JavaScript and Gateway activities the grammar within the Decision Flow must relate to Concept objects, not BOM objects, so as to be compliant with the Decision Graph execution environment.

At runtime the Decisions TEP operates as previously.

### Generic Business Studio Add-Ins

The generic Business Studio add-in operates as previously with modifications in line with the Decision Flow changes described above.

## Phase Four: AMX Decisions With Post-Deployment Rule Modification

### Generic AMX Decisions

*Note: this section describes the generic runtime features. There must also be a complimentary design time feature – refer to subsequent sections for more detail.*

In this phase the AMX Decisions SCA component is extended to include a new external management interface.

Minimal Business Events Engine

ADEC Wrapper

DEP

DEP

DEP

**AMX Decisions SCA Component**

Evaluate

Management

The management interface permits the contents of a Decision Table contained in a Decision Flow to be modified post-deployment providing the contract interface of the Decision Table does not change. This enables the logic in a Decision Table to be corrected or modified without having to redeploy the Decision Flow.

The ADEC wrapper updates the Decision Table definition via a read/write style interface.

### AMX BPM Specific

There are no changes to the AMX BPM specific architecture in this phase.

### Generic Business Studio Add-Ins

There are no changes to the generic Business Studio add-in architecture in this phase.

### Administration

This phase introduces an Administration interface that will either be web based or a derivative of BusinessEvents Decisions Manager. This will communicate with the AMX Decisions SCA component’s management interface in order to update Decision Table contents post-deployment.

# Mapping of Architectural Components to Use Cases

## Phase Two: (Variant One) AMX Decisions with Simple Decision Flow (AMX BPM)

### Deliverables

* Generic AMX Decisions SCA Component
* AMX BPM Decisions Task Extension Point for Process Engine
* AMX BPM Business Studio Decisions Add-Ins

### Deployment

The AMX Decisions SCA Component is deployed to any AMX node. The Decisions TEP must be deployed to the BPM node’s Process Engine.

The Business Studio Decisions add-ins are deployed to AMX BPM Business Studio.

### Installation

The Decisions SCA Component and Decisions Task Extension Point are packaged in an add-in installer for AMX. The Business Studio Decisions add-ins are packaged in an add-in installer for AMX BPM Business Studio. Both of these installers are executed at any point post-install and post-deployment of the BPM node and post-install of Studio.

## Phase Two: (Variant Two) AMX Decisions with Simple Decision Flow (non AMX BPM)

### Deliverables

* Generic AMX Decisions SCA Component
* Generic Business Studio Decisions Add-Ins

### Deployment

The AMX Decisions SCA Component can be deployed into any AMX node.

The Business Studio Decisions add-ins can be deployed to any compatible Business Studio but *not* AMX BPM Business Studio.

### Installation

The Decisions SCA Component is packaged in an add-in installer for AMX. The Business Studio Decisions add-ins are packaged in an add-in installer for Business Studio. Both of these installers are executed at any point post-install and post-deployment of AMX and post-install of Studio.

## Phase Three: AMX Decisions with Enhanced Decision Flow

### Deliverables

* Generic AMX Decisions SCA Component
* AMX BPM Decisions Task Extension Point for Process Engine
* AMX BPM Business Studio Decisions Add-Ins
* Generic Business Studio Decisions Add-Ins

### Deployment

The AMX Decisions SCA Component is deployed to any AMX node. The Decisions TEP must be deployed to the BPM node’s Process Engine.

The Business Studio Decisions add-ins are deployed to any compatible Studio instance.

### Installation

The Decisions SCA Component and Decisions Task Extension Point are packaged in an add-in installer for AMX. The Business Studio Decisions add-ins are packaged in an add-in installer for AMX (BPM) Business Studio. Both of these installers are executed at any point post-install and post-deployment of the node and post-install of Studio.

## Phase Four: AMX Decisions with Post-Deployment Rule Modification

### Deliverables

* Generic AMX Decisions SCA Component
* AMX BPM Decisions Task Extension Point for Process Engine
* AMX BPM Business Studio Decisions Add-Ins
* Generic Business Studio Decisions Add-Ins
* Administration GUI

### Deployment

The AMX Decisions SCA Component is deployed to any AMX node. The Decisions TEP must be deployed to the BPM node’s Process Engine.

The Business Studio Decisions add-ins are deployed to any compatible Studio instance.

The deployment of the Administration GUI is to be determined.

### Installation

The Decisions SCA Component and Decisions Task Extension Point are packaged in an add-in installer for AMX. The Business Studio Decisions add-ins are packaged in an add-in installer for AMX (BPM) Business Studio. Both of these installers are executed at any point post-install and post-deployment of the node and post-install of Studio.

The installation of the Administration GUI is to be determined.

# Dependencies and Requirements

The requirements for each phase are explicitly stated in full. Note that if a requirement from an earlier phase is *not* repeated in a later phase it is *not* required in that later phase.

## Phase Zero: A How to Guide Using Mediation

This has been successfully delivered using existing product.

## Phase One: A How to Guide Without Using Mediation

### Business Events

No new requirements.

### AMX Platform

Requires top level element support for Document/Literal web services.

### AMX BPM

Requires top level element support for Document/Literal web services.

### BPM Business Studio

Requires top level element support for Document/Literal web services for user defined BOM.

### BOM MetaData

Requires top level element support for Document/Literal web services for user defined BOM.

### Generic Business Studio

Not applicable for this phase.

### Administrator GUI

Not applicable for this phase.

## Phase Two: AMX Decisions with Simple Decision Flow

### Business Events

* Concept and Domain Model Generation API (Runtime)
* Concept construction/population/reading API (Runtime)
* Rule Function execution environment (Runtime)
* Packaging of subset of Eclipse into Business Studio (Design time)

### AMX Platform

* Packaging of a minimal Process Engine (Runtime)
* AMX Decisions Component (Runtime)

### AMX BPM

* Task Extension Point for Decision Service activity (Runtime)

### BPM Business Studio

* Decision Table, Concept and Domain model editor/viewer (Design time)
* Decision Service Activity (Design time)
* BOM to Concept and Domain Model generator (Design time)
* Simple Decision Flow Editor (Design time)
* Decision Flow to BE Rule Function generator (Design time)

### BOM MetaData

* Annotations (if necessary?)

### Generic Business Studio

* Decision Table, Concept and Domain model editor/viewer (Design time)
* BOM to Concept and Domain Model generator (Design time)
* Simple Decision Flow Editor (Design time)
* Decision Flow to BE Rule Function generator (Design time)

### Administrator GUI

* Not applicable for this phase.

## Phase Three: AMX Decisions with Enhanced Decision Flow

### Business Events

* Concept and Domain Model Generation API (Runtime)
* Concept construction/population/reading API (Runtime)
* XPDL to Decision Graph API (Runtime)
* Decision Graph execution environment (Runtime)
* Support for JavaScript within the Decision Graph environment (Runtime) – *To be clarified with the BusinessEvents team as a requirement.*
* Packaging of subset of Eclipse into Business Studio (Design time)

### AMX Platform

* Packaging of a minimal Process Engine (Runtime)
* AMX Decisions Component (Runtime)

### AMX BPM

* Task Extension Point for Decision Service activity (Runtime)

### BPM Business Studio

* Decision Table, Concept and Domain model editor/viewer (Design time)
* Decision Service Activity (Design time)
* BOM to Concept and Domain Model generator (Design time)
* Enhanced Decision Flow Editor (Design time)
* JavaScript and Gateway logic compatibility with Concept model (Design time) – *To be clarified with the BusinessEvents team as a requirement.*
* Support for Concept and Domain Model grammar in JavaScript and Gateway activities in the Decision Flow. (Design time) – *To be clarified with the BusinessEvents team as a requirement.*
* Decision Flow to BE XPDL converter (Design time)

### BOM MetaData

* Annotations (if necessary?)

### Generic Business Studio

* Decision Table, Concept and Domain model editor/viewer (Design time)
* BOM to Concept and Domain Model generator (Design time)
* Enhanced Decision Flow Editor (Design time)
* JavaScript and Gateway logic compatibility with Concept model (Design time) – *To be clarified with the BusinessEvents team as a requirement.*
* Support for Concept and Domain Model grammar in JavaScript and Gateway activities in the Decision Flow. (Design time) – *To be clarified with the BusinessEvents team as a requirement.*
* Decision Flow to BE XPDL converter (Design time)

### Administrator GUI

* Not applicable for this phase.

## Phase Four: AMX Decisions with Post-Deployment Rule Modification

### Business Events

* Concept and Domain Model Generation API (Runtime)
* Concept construction/population/reading API (Runtime)
* XPDL to Decision Graph API (Runtime)
* Decision Graph execution environment (Runtime)
* Packaging of subset of Eclipse into Business Studio (Design time)
* Rule modification API (Runtime)

### AMX Platform

* Packaging of a minimal Process Engine (Runtime)
* AMX Decisions Component (Runtime)
* Support for rule modification via AMX Decisions Component (Runtime)

### AMX BPM

* Task Extension Point for Decision Service activity (Runtime)

### BPM Business Studio

* Decision Table, Concept and Domain model editor/viewer (Design time)
* Decision Service Activity (Design time)
* BOM to Concept and Domain Model generator (Design time)
* Enhanced Decision Flow Editor (Design time)
* JavaScript and Gateway logic compatibility with Concept model (Design time) – *To be clarified with the BusinessEvents team as a requirement.*
* Decision Flow to BE XPDL converter (Design time)

### BOM MetaData

* Annotations (if necessary?)

### Generic Business Studio

* Decision Table, Concept and Domain model editor/viewer (Design time)
* BOM to Concept and Domain Model generator (Design time)
* Enhanced Decision Flow Editor (Design time)
* JavaScript and Gateway logic compatibility with Concept model (Design time) – *To be clarified with the BusinessEvents team as a requirement.*
* Decision Flow to BE XPDL converter (Design time)

### Administrator GUI

* Web or Decision Manager interface for rule modification (Run time)

# Accept 360 Requirements Reference

This section is yet to be updated

[To be updated and synchronised with the remainder of the document pending discussion with Roger]

The following table shows the referenced requirements in Accept 360 for each phase described in this document.

The status indicates whether the requirement requires:

* No change
* Modification to align with this document,
* A new requirement to be created,
* That the requirement already should exist in A360,
* The status is unknown in that it would exist outside of the BPM group.

|  |  |  |  |
| --- | --- | --- | --- |
| Reference | Requirement | Phase | Status |
| - | Phase 0 Use Case | 0 | New |
| 82883 | Phase 1 Use Case | 1 | Mod |
| 82939 | Package Feature | 1 | N/C |
| 82956 | Business Studio Installer, Design Time | 1 | N/C |
| 83014 | Installer, Runtime | 1 | N/C |
| 82966 | Decisions Perspective | 1 | N/C |
| 82960 | BOM to BE Concept Synchronisation | 1 | Mod |
| 82995 | Cross Project References | 1 | N/C |
| 82974 | Decision Service Project Type/Wizard | 1 | N/C |
| 82978 | Decision Service Activity | 1 | N/C |
| 82984 | Auto Map Parameters | 1 | N/C |
| 82991 | Tester/Debugger Integration | 1 | N/C |
| 83017 | DAA Builder for Decision Service | 1 | N/C |
| 83022 | Rule Loader Integration | 1 | N/C |
| 83032 | Decisions Implementation Type | 1 | N/C |
|  | BE Concept Generation API for Studio | 1 | Unknown |

# Appendix A: Glossary

|  |  |
| --- | --- |
| Term | Definition |
| Concept Model | An abstract entity similar to the object-oriented concept of a class. A concept is a description of a set of properties that, when grouped together, create a meaningful unit. Concepts can be organized in a hierarchical structure. |
| Domain Model | A domain model specifies the values that you may find useful for defining ontology item properties. For example, instead of typing text for a certain concept property, you can pick a value from a list, or enter a value within a predefined range. |
| Decision Table | A tabular form presenting a set of conditions and their corresponding actions. A graphical tool for building rules. |
| Virtual Rule Function | A rule function whose signature is defined in a BusinessEvents project and whose implementation is defined using decision tables. |