WSDM Management of Web Services Requirements

Working Draft July 2003

Document identifier:
wsdm-mows-requirements-20030727

Location:
http://www.oasis-open.org/?

Editors:
Ellen Stokes, IBM, <stokese@us.ibm.com>
Heather Kreger, IBM, <kreger@us.ibm.com>
Larry Lackey, TIBCO <llackey@tibco.com>
Veena Subrahmanyam, Hewlett-Packard <veena.subrahmanyam@hp.com>
Sanjeev Kumar, Individual <sakumar@attbi.com>

Abstract:
This document is the requirements for the Management of Web Services specification of the Web Services Distributed Management Technical Committee.

Status:
This document is a **working draft** of the OASIS Web Services Distributed Management (WSDM) Technical Committee. We solicit your comments.

Committee members should send comments on this specification to the wsdm@lists.oasis-open.org list. Others should subscribe to and send comments to the wsdm-comment@lists.oasis-open.org list. To subscribe, send an email message to wsdm-comment-request@lists.oasis-open.org with the word "subscribe" as the body of the message.

For information on whether any patents have been disclosed that may be essential to implementing this specification, and any offers of patent licensing terms, please refer to the Intellectual Property Rights section of the WSDM TC web page (http://www.oasis-open.org/committees/wsdm/).

Table of Contents

1 Introduction

1.1 Purpose

This document is the requirements for the Management of Web Services specification of the Web Services Distributed Management Technical Committee, whose purpose and deliverables (http://www.oasis-open.org/committees/wsdm/charter.php) are.

Statement of Purpose

To define web services management. This includes using web services architecture and technology to manage distributed resources. This TC will also develop the model of a web service as a manageable resource. This TC will collaborate with various evolving activities within other standards groups, including, but not limited to, DMTF (working with its technical work groups regarding relevant CIM Schema), GGF (on the OGSA common resource model and OGSI regarding infrastructure), and W3C (the web services architecture committee). Also liaison with other OASIS TCs, including the security TC and other management oriented TCs.

List of Deliverables

Web Services Distributed Management (WSDM) V1.0 Specification, Jan 2004 this includes WSDL described manageable resources and the xml schema to complete those descriptions. This document will also define explicit manageability for the components of the Web Services Architecture (WSA) as defined by the W3C.

A companion document of the Committee is the requirements for Management Using Web Services **[MUWS]**.

1.2 Terminology

The key words MUST, MUST NOT, REQUIRED, SHALL, SHALL NOT, SHOULD, SHOULD NOT, RECOMMENDED, MAY, and OPTIONAL in this document are to be interpreted as described in **[RFC2119]**.

2 Requirements

Management of Web Services MUST enable the management and provisioning of Web Services.

2.1 Manageability Model Requirements

2.1.1 Relationship

- 2 levels of relationships, svc types, endpoint refs [114]
- Manageability model include capabilities to identify, meter, monitor, conf, control, relate manageable elements [104]
- Model able to represent relationships between architectural elements and functional elements of Web service itself [103]
- Relationships [2]
- Supports the representation of Relationships between Web services and Web services and other IT resources [120]
- Understanding side effects involving other services, requirements to exist, dependencies, performance [114]
- Relationships to other Web services and other IT resources, things [34]

2.1.2 Metrics

- Ability to define and track metrics [22]
- Manageability model include capabilities to identify, meter, monitor, conf, control, relate manageable elements [104]
- Measure response time per request [46]
- Measurable to operation level [75]
- Monitoring of service – query avail, query perf, generic extensions [96]
- Define key performance metrics and configuration metrics, ability to change running environment [33]
- If a request involves other sub-operations get time taken in sub-operations [47]
- Health and performance monitoring, including pings, events, resources consumed by each request and status of each request [91]
- Diagnostic capabilities [82]

2.1.3 Model

- Ability to have consumer as well as provider perspective of service [86]
- Built on extensions to models, such as CIM [135]
- Core management info model required of every Web service [62]
- Core set management operations [64]
- Data model to capture meta-model [1]
- Define Group, categorization, types of Web services [36]

- Defined as extension to existing model… standard set of attributes, operations, extensibility [40]
- Event reporting of normal and abnormal [81]
- Manageability information model agnostic of how the information is represented or accessed [118]
- Mgmt (data) and admin (metadata) [13]
- Mgmt leveragable at an operations and business level [108]
- Model neutrality, not necessarily one method to model environment [67]
- Scalability of the model [68]
- Support multilevel granularity: compose-ability, extensibility [133]
- Supports the following manageability capabilities as defined by W3C WS Architecture Management Task Force Web Service Endpoint draft (http://www…) Identity , Configuration, Metrics, State, Operations, Events [119]
- What management things that are specific to web services that are new management information – i.e. standard events (for correlation), types of events that allow mgmt app to do further diagnosis or corrective action [5]
- Based on clearly defined model for management information, where meta-information of the model is available to manageability and management side of the model [142]
- Ability to monitor and control state via push events and pull state [11]
- Enable manageability (not just apps thru which it travels, built into resource itself too. Be able to look at status/state and know how its doing) [69]
- Where – where is it located, are the services restricted, dependencies, what are the endpoint specifics and access information [127]
- WS-Policy, WS-Policy Attachment, security policy, management is another policy that can be plugged into the WS-Policy framework. WS-Management policy. There exists a framework to associate policies with wsld and UDDI [54]

2.1.4 Lifecycle / State

- Ability to track status, health, degrees of up [32]
- Lifecycle state machine [92]
- Lifecycle: current state of svc – observed and reported [29]
- Long running and transient [145]
- Manageability model include capabilities to identify, meter, monitor, conf, control, relate manageable elements [104]
- Monitoring of service – query avail, query performance, generic extensions [96]
- Ping a Web service without doing a real operation or changing its state [45]
- Separation of status and state [146]
- Service operations –start, stop, ping [77]
- Set of operations to stop… variations – stop immediately, stop quiesce [138]
- Status type of unknown [143]
- What is ping in this context (container, application, …) [59]
- When – state, enabled, disabled, administratively affect this, errors and events too [128]
- Lifecycle mgmt [10]

- Lifecycle mgmt – deploy, start, stop, check dependencies, configuration mgmt, undeploy [90]
- Ability to control lifecycle and monitor [23]
- Control of service [94]

2.1.5 Types of Manageable Resources

- Business process instances behind a Web service (who requested, when, what steps, what resources other than Web service and non-Web services are required) <“should be able to model bus process instances as a manageable element”> [84]
- Define model for Web service execution environment [35]
- Long running and transient [145]
- Manage different scopes of services: individual, composite, process based [109]
- Management of conversations [147]
- Management of Web services means managing the Web service resource as an IT resource. It does not imply the ability to determine and manage the components that are used to implement the Web service [117]
- Stateless / stateful [144]
- Addresses the management of the Web services in the context of the Web services architecture, this may require being able to manage the role and management requirements of: Web Service Execution Environment, Discovery agency [122]

2.1.6 Identification

- WS-Addressing allows an ID and relationship defined; ID of endpoints should use this [55]
- Manageability model include capabilities to identify, meter, monitor, conf, control, relate manageable elements [104]
- Versioning [38]
- Who – identity, backing software behind it to validate version levels, etc [125]
- Unique handle / URI for a Web service – interoperable across vendors [31]
- Ability to determine service identity or quality – multiple URIs for same service instance, multiple svc instances w/ shared or identical state w/ shared semantics. [88]

2.1.7 Extensibility

- Monitoring of service – query avail, query performance, generic extensions [96]
- Disabling of monitoring by service itself [95]
- What – discovery of function provided, mandatory vs. optional [126]
- Consistency requirements – more than order to start/monitor services based on cross relationships w/ multiple services to (consistency snap shot?) [117]
- Change mgmt, another service could subscribe and be notified when a service has changed [8]

2.1.8 Configuration

- Manageability model include capabilities to identify, meter, monitor, conf, control, relate manageable elements [104]
- Service configuration – how generic? [78]

2.1.9 Policy and Orchestration

- Orchestration, choreography, business process issues, managing a larger unit and implications of it [21]
- Policy – what actions you take when you get a given event, state, quality of service, avail, *ilities, performance [18]
- WS orchestration [37]

2.2 Web Services Infrastructure

- [I.1] delegated support [115]
    - [I.1.1] federation of managers: permits multiple managers can get info from on agent [59]
- [I.2] federated support [115]
- [I.3] Efficiency – scalability, support for lightweight impl in resource constrained env [79]
- [I.4] scalability [79, 132]
- [I.5] enable operations to be reliable [139]
- [I.6] enables operation in occasionally connected environments [141]
- [1.7] Based on latest WS standards [134, 42, 116]
- [I.8] enables asynchronous and synchronous operations [87]
- [I.9] defines manageability capability profiles [61]
- [I.10] supports usecases for: [107]
    - [I.10.1] outsource mgmt
    - [I.10.2] hosted service
    - [I.10.3] grid
- [1.11] enable ordering of event notifications from a service [60]
- [I.12] Defined consistently (taking into consideration) existing management specifications, especially DMTF, and GGF [131]
- [I.13] enables propagation of management context between WS and WS environments [44]
- [I.14] support one operation on sets of Web services [15]
    - [1.14.1] in sequence [15]
    - [1.14.2] return groups of responses [15]
- [I.15] support many operations on one WS [15]
- [I.16] enables management of resources that span multiple Web services [26]
    - [I.16.1] session (WS-Conversation)
    - [I.16.2] transaction (WS-Transaction)
- *** Remote/local issues (“Transparency and its place”), affinity, security, etc. <TC Help> [16]

2.3 Secure

- [S.1] Must be able to be secure for authorized access [136]
- [S.2] Must support authentication between managed resources and managers [57]
  - [S.2.1] enables authorization of discovery [43]
- [S.3] Must support secure channels [74]
  - [S.3.1] encryption
  - [S.3.2] message integrity [80]

2.4 Interoperability

- [Interop.1] enable access to and discovery of manageability in a standard interoperable form – web services, description has to fit in whatever mechanisms used to describe ws <dup of 'using WS'?> [101]
- [Interop.2] work and use other ws standards, i.e. orchestration, choreography, transaction, if can't support those standards and manage their needs and at their business level won't be effective.

  [Interop.2.1] don't do things to shut of future standards; future proofing, <dupl. Of Extensibility> [76]

- [Interop.3] Mgmt capabilities Exposed considering WS-I basic profile [106]

2.5 Access to Model

- [access.1] access is decoupled from provider of manageability [102]
    - [access.1.1] supports direct access to resources [70]
    - [access.1.2] supports access to resources through agents [70]
- [access.2] access is decoupled from discovery of manageability information [102]
- [access.3] access is decoupled from the model and semantics of manageability [100, 102]
- [access.4] Must be defined consistent with “management using web services” [121]
    - [access.4.1] canonical representation
    - [access.4.2] discovery
    - [access.4.3] access [71]
    - [access.4.4] accessible through firewalls [71]
    - [access.4.5] accessible using protocols in addition to http. [56]
- [access.5] Manageability model represented as a describable interface [98]
- [access.6] enable access to manageability information in multiple management domains [73]
    - [access.6.1] in support of WS-transactions or WS-conversations across federated domains. [73]
- [access.7] must support introspection of methods and state [12]
    - [access.7.1] from the resources description [12]
    - [access.7.2] from the resource itself [12]

2.6 Extensibility

- [Ext.1] Extensibility for resource customization – allows additional, custom, service specific manageability capabilities to be added. [124, 63, 66, 105, 130]
    - [ext.1.1] modeled
    - [ext.1.2] discovered
    - [ext.1.3] accessed
- [Ext.2] Extensibility for future-proofing: extensible solution because we know we will not completely satisfy future requirements [4]
- [Ext.3] Additional manageable roles may be identified and defined. [123]

2.7 Management Application / Usage Scenarios Enabled

- [Scenarios.1] Enables association of policy with services and service groups [27]
- [Scenarios.2] enables accounting [19, 39]
- [scenarios.3] enables billing [19, 39]
- [scenarios.4] enables metering [19, 25, 39]
- [scenarios.5] enables auditing [19, 39]
- [scenarios.6] enables performance assessment per hop [52]
- [scenarios.7] enables business impact analysis for a service [111]
- [scenarios.8] enables management of business processes [112]
- [scenarios.9] enables the use of business processes for resource management [112]
- [scenarios.10] enables end to end management of web services and underlying infrastructure [51]
- [scenarios.11] enables provisioning management [48, 20]
- [scenarios.12] enables quality of service management [48]
- [scenarios.13] enables transaction performance management [48]
- [scenarios.14] enables operations management [48]
- [scenarios.15] enables service level agreement management [48]
- [scenarios.16] enables management of legacy applications? [48]
- [scenarios.17] enables management of new applications? [48]
- [scenarios.18] enables grouping of services into arbitrary groups (aggregations) for mgmt purposes [30]
- [scenarios.19] enables assessment and quantification of health of a web service [7]
- [scenarios.20] enables manual and automatics operations [85]
- [scenarios.21] enables deployment management [93]
- [scenarios.22] enables lifecycle management [50]
- [scenarios.23] enables monitoring [50]
- [scenarios.24] enables aggregation of metrics and configuration [53]
- [scenarios.25] enables root cause analysis [110]
- [scenarios.26] enables problem diagnosis [6]
- ***Message tracking, be able to track messages that have been misrouted <TC Help, what is this> [9]

2.8 Miscellaneous

- [misc.1] describe the manageability capabilities we are trying to satisfy [3]
- [misc.2] consider specifications which are not committed standards [140]
- [misc.3] describe business value provided by web services manageability [83, 97, 129]

2.9 Discoverabliity

- [discov.1] enables ability to discover configuration and configuration changes [28]
- [discov.2] enables ability to discovery management capabilities [24,89]
- [discov.3] enable ability to discover manageable web services via the same mechanisms as other web services [41, 89]

Usability

- adoptable, easy to develop, developer friendly, adopted in a gradual way, allow partial support of capabilities, and then build on top of that [72, 49]
    - [usability.1] developer friendly
        - [usability.1.1] easy to understand
        - [usability.1.2] easy to develop
        - [usability.1.3] supports multiple programming environments
    - [usability.2] incrementally implementable
        - [usability.2.1] by service developer
- [usability.3] Low impact on implementor of manageability [148, 99, 113]
- [usabilty.4] Minimum overhead added to service [137]
- [usability.5] specification provides clear guidance to developers using reference implementations, toolkits, and internal parts to instrument [65]

3 Use Cases

3.1 Web Services Endpoint dependent upon other Web Services Endpoints

3.2 Web Services Endpoint dependent on other IT Resources

4 References

4.1 Normative

[RFC2119] S. Bradner, *Key words for use in RFCs to Indicate Requirement Levels*, http://www.ietf.org/rfc/rfc2119.txt, IETF RFC 2119, March 1997.

[MUWS] ?, ?, http://www.oasis-open.org/?, OASIS WSDM TC, ? 2003.

4.2 Non-Normative

[WSAR] A. Austin, A. Barbir, C. Ferris, S. Garg, *Web Services Architecture Requirement*, http://www.w3.org/TR/wsa-reqs, W3C, November 2002.

Appendix A. Acknowledgments

The editors would like to acknowledge the contributions of the OASIS Web Services Distributed Management Technical Committee, whose voting members at the time of publication were:

- A. Nonymous (chair), Example Corp.
-

Appendix B. Brainstorming

TC Mission

3. Clear understanding of the manageability capabilities we're trying to satisfy

A. Manageability Model Requirements

1. Datamodel to capture metadata

2. Relationships

5. What management things that are specific to web services that are new management information – i.e. standard events (for correlation), types of events that allow mgmt app to do further diagnosis or corrective action.

7. Health rating used by other apps. Web service would decide what its health rating so a client can figure out which instance is used, displayed by mgmt app

9. Message tracking, be able to track messages that have been misrounted

10. Lifecycle mgmt

11. Ability to monitor and control state via push events and pull state

12. Introspection of methods and state

13. Mgmt (data) and admin (metadata)

14. Understanding side effects involving other services, requirements to exist, dependencies, and performance

17. Consistency requirements – more than order to start/monitor services based on cross relationships w/ multiple svcs to (consistency snap shot?)

18. Policy – what actions you take when you get a given event, state, qos, avail, *ilities, perf,

21. Orchestration, choreography, business process issues, managing a larger unit and implications of it

22. Ability to define and track metrics

23. Ability to control lifecycle and monitor

24. Abilty to find out mgmt capabilities available

25. Find usage of service (who doing what)

26. Ability to track things larger than a request… session, transaction, …

27. Ability to assoc mgmt rules with a service or collection of services

28. Ability to find out and track general configuration

29. Lifecycle: current state of svc – observed and reported

30. Grouping of services into arbitrary groups (aggregations) for mgmt purposes

32. Tracking status, health, degrees of upnessness

34. Define key perf metrics and config metrics, ability to change running environ

35. Relationships to other ws and other IT resources, things

36. Define ws execution environment

37. Group, categorization, types of webservices

38. Ws orchestration

39. Versioning

41. Defined as extension to existing model… standard set of attributes, operations, extensibility

45. Need a way to carry and propogate mgmt tags across ws and ws env. Challenge today for creating units of work is there's no way to propogate info independently of the service itself

B. Web Services Infrastructure

31. Unique handle/uri for a webservice – interoperable across vendors

30. Grouping of services into arbitrary groups (aggregations) for mgmt purposes

42. Discoverable via ws,

43. Soapy/wsdl

45. Need a way to carry and propogate mgmt tags across ws and ws env. Challenge today for creating units of work is there's no way to propogate info independently of the service itself

C. Secure

44. Secure, not open to unauth usage or discovery

D. Interoperability

31. Unique handle/uri for a webservice – interoperable across vendors

E. Access to Model

15. Bulk operations – set, sequences, partial orders

16. Remote/local issues ("Transparency and its place"), affinity, security, etc.

30. Grouping of services into arbitrary groups (aggregations) for mgmt purposes

F. Extensibility

4. Extensible solution because we know we will not completely satisfy future requirements

G. Management Application/Usage Scenarios enabled

6. Standard for diagnosis

7. Health rating used by other apps. Web service would decide what its health rating so a client can figure out which instance is used, displayed by mgmt app

8. Change mgmt, another service could subscribe and be notified when a service has changed

9. Message tracking, be able to track messages that have been misrounted

18. Policy – what actions you take when you get a given event, state, qos, avail, *ilities, perf,

19. Accounting, billing, metering, auditing

20. Provisioning (data and code)

21. Orchestration, choreography, business process issues, managing a larger unit and implications of it

40. Billing, auditing

448 Raw list:

449 A. Manageability Model Requirements

(A) 2 levels of relationships, svc types, endpoint refs (Relationship)	114.
(A) Ability to define and track metrics (metrics)	22.
(A) Ability to have consumer as well as provider perspective of service (model)	86.
(A) Ability to track status, health, degrees of upnessness (lifecycle/state)	32.
(A) Built on extenstions to models, such as cim (model)	135.
(A) Business process instances behind a webservice (who requested, when, what steps, what resources other ws and non-ws are required) <“should be able to model bus process instances as a manageable element”> (types of manageable resources)	84.
(A) core management info model required of every ws (model)	62.
(A) core set mgmt ops (model)	64.
(A) Datamodel to capture metadata (model)	1.
(A) Define Group, categorization, types of webservices (model)	36.
(A) Define model for ws execution environment (types of manageable resources)	35.
(A) Defined as extension to existing model… standard set of attributes, operations, extensibility (model)	40.
(A) Event reporting of normal and abnormal (model)	81.
(A) for an id, ws-addressing allows and id and relationship def., id of endpoints should use this (identification)	55.
(A) Lifecycle state machine (lifecycle/state)	92.
(A) Lifecycle: current state of svc – observed and reported (lifecycle/state)	29.
(A) Long running and transient	145.

(lifecycle/state, types of managed resources)	
(A) Manage diff scopes of svcs: individual, composite, process based (types of managed resources)	109.
(A) Manageability information model agnostic of how the information is represented or accessed (model)	118.
Manageability model include capabilities to identify, meter, monitor, conf, control, relate manageable elements (identification, metrics, configure, lifecycle/state, relationships)	104.
(A) Management of conversations (types of managed resources)	147.
(A) Management of Web services means managing the Web service resource as an IT resource. It does not imply the ability to determine and manage the components that are used to implement the Web service. (types of managed resources)	117.
(A) Measure response time per request (metric)	46.
(A) measureable to operation level (metric)	75.
(A) Mgmt (data) and admin (metadata) <(model)???>	13.
(A) Mgmt leveragable at an operations and business level (model)	108.
(A) Model able to represent relationships between arch elements and funct elements of ws itself (relationships)	103.
(A) model neutrality, not necessarily one method to model env (model)	67.
(A) Monitoring of service – query avail, query perf, generic extensions (metric, lifecycle/state, extensibility)	96.
(A) Ping a ws w/o doing a real operation or changing its state (lifecycle/state)	45.
(A) Relationships (relationships)	2.
(A) scalability of the model (model)	68.
(A) Separation of status and state (lifecycle/state)	146.

(A) Service config – how generic? (configuration)	78.
(A) Service operations –start,stop, ping (lifecycle/state)	77.
(A) Set of operations to stop… variations – stop immed, stop quiese, (lifecycle/state)	138.
(A) Stateless/statefull (types of manageable resources)	144.
(A) Status type of unknown (lifecycle/state)	143.
(A) Support multilevel granularity (model) <composability, extensibility>	133.
(A) Supports the following manageability capabilities as defined by W3C WS Architecture Management Task Force Web Service Endpoint draft (http://www…) Identity , Configuration, Metrics, State, Operations, Events (model)	119.
(A) Supports the representation of Relationships between Web services and Web services and other IT resources. (relationships)	120.
(A) Understanding side effects involving other services, requirements to exist, dependencies, performance (relationships)	14.
(A) Versioning (identification)	38.
(A) what is ping in this context (container, application, …) (lifecycle/state)	59.
(A) What management things that are specific to web services that are new management information – i.e. standard events (for correlation), types of events that allow mgmt app to do further diagnosis or corrective action. (model)	5.
(A) When – state, enabled, disabled, administratively affect this, errors and events too (lifecycle/state)	128.
(A) Who – identity, backing software behind it to validate vers levels, etc. (identfication)	125.
(A)Addresses the management of the Web	122.

services in the context of the Web services architecture, this may require being able to manage the role and management requirements of: Web Service Execution Environment, Discovery agency , (types of manageable resources)	
(A, B) Wspolicy, wsp-attachment, security policy, management is another policy that can be plugged into the wspolicy framework. Wsmanagement policy. There exists a fw to associate policies w/ wsld and uddi, (	54.
(A,B) Ability to determine service identity or quality – multiple uri's for same service instance, multiple svc instances w/ shared or identical state w/ shared semantics.	88.
(A,D) Relationships to other ws and other IT resources, things	34.
(A,D) Unique handle/uri for a webservice – interoperable across vendors	31.
(A,E) Ability to monitor and control state via push events and pull state	11.
(A,E) Based on clearly defined model for management information, where metainformation of the model is available to manageability and management side of the model	142.
(A,E) enable manageability (not just apps thru which it travels, built into resource itself too. Be able to look at status/state and know how its doing)	69.
(A,E) Where – where is it located, are the services restricted, dependencies, what are the endpoint specifics and access information	127.
(A,G) Ability to control lifecycle and monitor	23.
(A,G) Change mgmt, another service could subscribe and be notified when a service has changed	8.
(A,G) Consistency requirements – more than order to start/monitor services based on cross relationships w/ multiple svcs to (consistency snap shot?)	17.
(A,G) Control of service (lifecycle mgmt)	94.

(A,G) Define key perf metrics and config metrics, ability to change running environ	33.
(A,G) Diagnostic capabilities	82.
(A,G) Disabling of monitoring by service itself	95.
(A,G) Health and perf monitoring, including pings, events, resources consumed by each request and status of each request	91.
(A,G) If a request involves other suboperations get time taken in suboperations	47.
(A,G) Lifecycle mgmt	10.
(A,G) Lifecycle mgmt – deploy, start, stop, check dependencies, config mgmt, undeploy	90.
(A,G) Orchestration, choreography, business process issues, managing a larger unit and implications of it	21.
(A,G) Policy – what actions you take when you get a given event, state, qos, avail, *ilities, perf,	18.
(A,G) Ws orchestration	37.
(A,I) What – discovery of funct provided, mandatory vs. optional	126.
B. Web Services Infrastructure	
(B) [I.1] delegated support [I.2] federated support	115.
(B) [I.3] Efficiency – scalability, support for lightweight impl in resource constrained env [I.4] scalability	79.
(B) [I.1.1] federation of managers: permits multiple managers can get info from on agent	58.
(B) logically order of events sent to manager [I.4] enable ordering of event notifications from a service	60.
(B) Management operations are reliable [I.5] enable operations to be reliable	139.
(B) REALLY distributed (occasionally	141.

connected, not always connected, discontinuous) [I.6] enables operation in occasionally connected environments	
(B) Remote/local issues (“Transparency and its place”), affinity, security, etc. <TC Help>	16.
(B) Scalability to LARGE distributed systems <dup of I.4>	132.
(B) Soapy/wsdl <dup of I.7>	42.
(B) Support for asynch and deferred reply service use models, whether bus process behind it or not [I.8] enables asynchronous and synchronous operations	87.
(B,C) need predefined roles of capabilities [I.9] defines manageability capability profiles	61.
(B,D) Based on latest ws standards [I.7]	134.
(B,D) Consideration for all ws env. , i.e. hosted svcs for outsource mgmt, grid [I.10] supports usecases for: [I.10.1] outsource mgmt [I.10.2] hosted service [I.10.3] grid	107.
(B,D) [I.12] Defined consistently (taking into consideration) existing management specifications, esp DMTF, and GGF	131.
(B,D) leverage and consistent w/ existing standards <dup of 134>	116.
(B,D) Need a way to carry and propagate mgmt tags across ws and ws env. Challenge today for creating units of work is there’s no way to propogate info independently of the service itself [I.13] enables propogation of management context between WS and WS environments	44.
(B,E) Bulk operations – set, sequences, partial orders [I.14] support one operation on sets of WSs [1.14.1] in sequence [1.14.2] return groups of responses [I.15] support many operations on one WS	15.

(B,G) Ability to track things larger than a request… session, transaction, … <TC help> [I.15] enables management of resources that span multiple Web services [I.15.1] session (WS-Conversation) [I.15.2] transaction (WS-Transaction)	26.
C. Secure	
(C) [S.1] Must be able to be secure for authorized access	136.
(C) prevent hijack of agents by rogue managers – handshake/auth [S.2] Must support authentication between managed resources and managers	57.
(C) secure at communication and operational level, [S.3] Must support secure channels [S.3.1]encryption	74.
(C) Secure, not open to unauth usage or discovery [S.2.1] enables authorization of discovery	43.
(C) Security – authorization and msg integrity [S.3.2] message integrity	80.
D. Interoperability	
(D) [Interop.1] enable access to and discovery of manageability in a standard interoperable form – web services, description has to fit in whatever mechanisms used to describe ws <dup of 'using WS'?>	101.
(D) [Interop.2] work and use other ws standards, i.e. orchestration, choreography, transaction, if can't support those standards and manage their needs and at their business level won't be effective. [Interop.2.1] don't do things to shut of future standards; future proofing, <dupl. Of Extensibility>	76.
(D,E) [Interop.3] Mgmt capabilities Exposed considering ws-I basic profile	106.

E. Access to Model	
(E) [access.1] Access and discovery and model independent of provider of manageability (service, env, 3rd party) [access.1] access is decoupled from provider of manageability [access.2] access is decoupled from discovery of manageability information	102.
(E) Access and mechanics separate from model and semantics of manageability [access.3] access is decoupled from the model and semantics of manageability <100, 102>	100.
(E) [] Canonical representation, discovery, and access consistent with "Management Using Web services" is defined. [access.4] Must be defined consistent with "management using web services" [access.4.1] canonical represenation [access.4.2] discovery [access.4.3] access [access.4.4] accessible through firewalls <71> [access.4.5] accessible using protocols in addition to http. <56>	121.
(E) firewall friendly <to access.4.4>	71.
(E) [access.5] Manageability model represented as a describable interface	98.
(E) protocol agnostic access to management data - not dependent on http <to access.4.4.5>	56.
(E) support federated domains, federated management, esp w/ ws because its used in trans type env. And to do end to end need to account for fact that trans is spread across mgmt domains and need to build full view [access.6] enable access to manageability information in multiple management	73.

domains [access.6.1] in support of ws-transactions or ws-conversations across federated domains.	
(E) support various modes of deployment, agent based or agentless, <to access.1> [access.1.1] supports direct access to resources [access.1.2] supports access to resources through agents	70.
(E,I) [access.7] must support introspection of methods and state [access.7.1] from the resources description [access.7.2] from the resource itself	12.
F. Extensibility	
(F) Extensibility <dup of 124>	105.
(F) [Ext.1] Extensibility for resource customization – allows additional, custom, service specific manageability capabilities to be added. [ext.1.1] modeled [ext.1.2] discovered [ext.1.3] accessed	124.
(F) extensibility of model <dup of 66>	63.
(F) extensibility of model and capabilities <dup of 124>	66.
(F) Extensible <dup of 124>	130.
(F) [Ext.2] Extensibility for future-proofing: extensible solution because we know we will not completely satisfy future requirements	4.
(F) [Ext.3] Additional manageable roles may be identified and defined.	123.
G. Management Application/Usage Scenarios enabled	
(G) Ability to assoc mgmt rules with a service or collection of services [Scenarios.1] Enables association of policy	27.

with services and service groups	
(G) Accounting, billing, metering, auditing [Scenarios.2] enables accounting [scenarios.3] enables billing [scenarios.4] enables metering [scenarios.5] enables auditing	19.
(G) Assess perf and turnaround at every hop [scenarios.6] enables performance assessment per hop	52.
(G) Billing, auditing <dup 19>	39.
(G) Business impact at svc level [scenarios.7] enables business impact analysis for a service	111.
(G) Consideration for process driven approach to mgmt [scenarios.8] enables management of business processes [scenarios.9] enables the use of business processes for resource management	112.
(G) Enable transaction perf, ops, provisioning, qos, controls <dup of ops>, end to end <dup of 51>, sla, legacy apps, new apps [scenarios.11] enables provisioning management [scenarios.12] enables quality of service management [scenarios.13] enables transaction performance management [scenarios.14] enables operations management [scenarios.15] enables service level agreement management [scenarios.16] enables management of legacy applications? [scenarios.17] enables management of new applications?	48.
(G) End to end mgmt of ws and underlying infrastructure [scenarios.10] enables end to end management of web services and underlying infrastructure	51.
(G) Find usage of service (who doing what)	25.

<dup of scenarios.4>	
(G) [scenarios.18] enables grouping of services into arbitrary groups (aggregations) for mgmt purposes	30.
(G) Health rating used by other apps. Web service would decide what its health rating so a client can figure out which instance is used, displayed by mgmt app [scenarios.19] enables assessment and quantification of health of a web service	7.
(G) How can someone act on it manually or automatically (even w/in process) [scenarios.20] enables manual and automatics operations	85.
(G) Integration with deployment process [scenarios.21] enables deployment management	93.
(G) Lifecycle mgmt/monitoring [scenarios.22] enables lifecycle management [scenarios.23] enables monitoring	50.
(G) Message tracking, be able to track messages that have been misrouted <TC Help, what is this>	9.
(G) Perf and config metrics for ws and infra to be rolled up to whatever level of abstraction user wants it at [scenarios.24] enables aggregation of metrics and configuration	53.
(G) Provisioning (data and code) <dup of scenarios.11>	20.
(G) Root cause analysis at service level [scenarios.25] enables root cause analysis	110.
(G) Standard for diagnosis [scenarios.26] enables problem diagnosis	6.
H. Miscellaneous	
(H) Clear understanding of the manageability capabilities we're trying to satisfy [] describe the manageability capabilities we are trying to satisfy	3.

(H) Consider non-standard specs (not committed vs not never) [] consider specifications which are not committed standards.	140.
(H) Intent of this – manageability has to carry sufficient capabilities to be able to do … what? Why is it valuable <dup 83>	97.
(H) Why – this is what we do <dup 83>	129.
(H) Why business users would care about this – whats the value we bring [] describe business value provided by web services manageability	83.
I. Discoverability	
Ability to find out and track general configuration [discov.1] enables ability to discover configuration and configuration changes	28.
Ability to find out mgmt capabilities available [discov.2] enables ability to discovery management capabilities	24.
Discoverable via ws, [discov.3] enable ability to discover manageable web services via the same mechanisms as other web services	41.
(I) Discovery – finding ws, introspection <dup 41 and 24>	89.
J. Usability	
(J) adoptable, easy to develop, developer friendly, adopted in a gradual way, allow partial support of capabilities, and then build on top of that [usability.1] developer friendly [usability.1.1] easy to understand [usability.1.2] easy to develop [usability.1.3] supports multiple programming environmens <49> [usability.2] incrementally implementable [usability.2.1] by service developer	72.

(J) [usability.5] specification provides clear guidance to developers using reference implementations, toolkits, and internal parts to instrument	65.
(J) Developer – easy to instrument by app vendor, low barrier to entry, support multiple programming environments ,dup of 72>	49.
(J) Implementation burden on manager not developer [usability.3] Low impact on implementor of manageability <148>	148.
(J) Low impact on implementer of manageability …more work on manager than exposer of interface <dup of 148>	99.
(J) Lower incr impact to svc developer <dup of 148>	113.
(J) [usabilty.4] Minimum overhead added to service	137.

450

451

Appendix C. Notices

OASIS takes no position regarding the validity or scope of any intellectual property or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; neither does it represent that it has made any effort to identify any such rights. Information on OASIS's procedures with respect to rights in OASIS specifications can be found at the OASIS website. Copies of claims of rights made available for publication and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementors or users of this specification, can be obtained from the OASIS Executive Director.

OASIS invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights which may cover technology that may be required to implement this specification. Please address the information to the OASIS Executive Director.

Appendix D. Revision History

Rev	Date	By Whom	What
2003-05	2003-05	Editors	Initial version
2003-0727	2003-05-27	Ellen Stokes	Put Brainstorming section and table into text (chapter 2)