SDPng
Model and Strawman Syntax

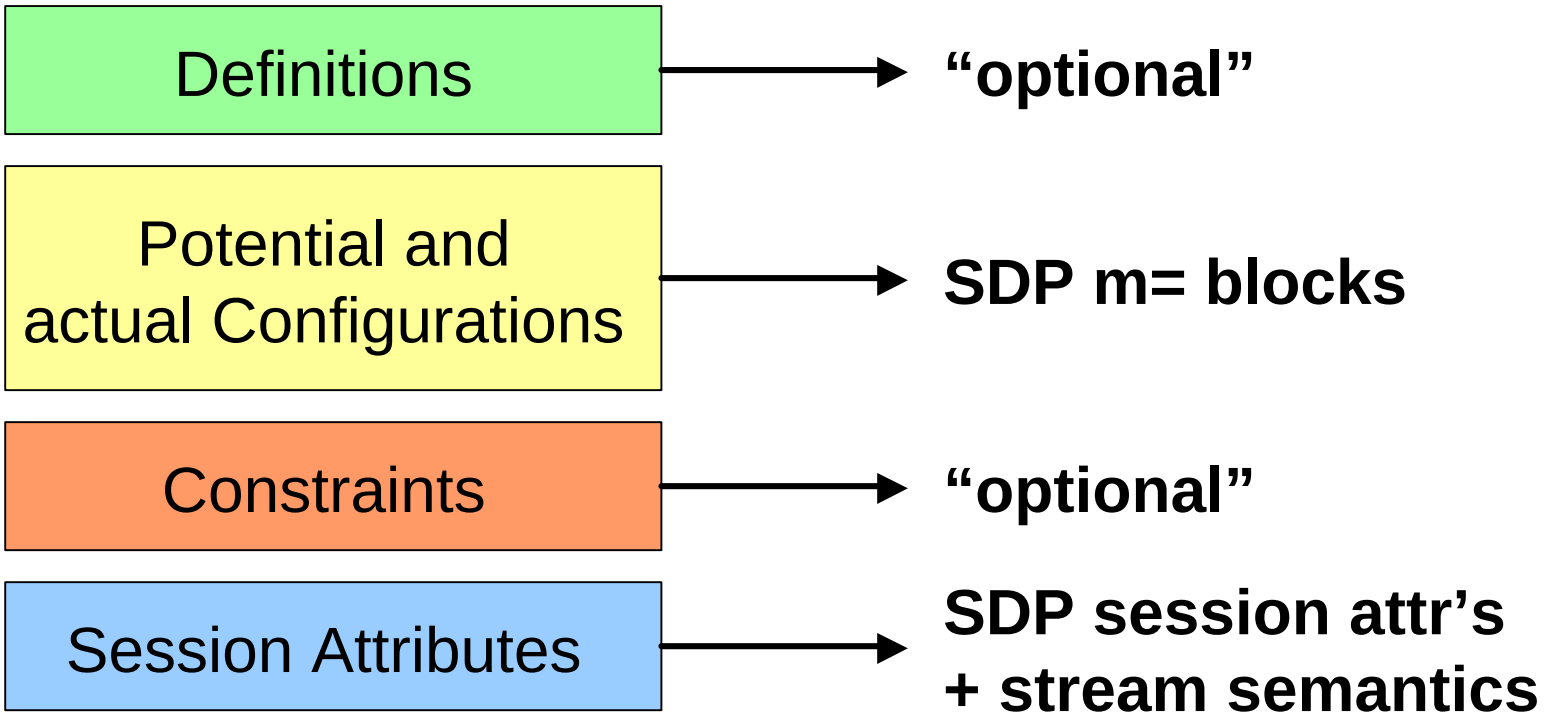
Dirk Kutscher	dku@tzi.org
Jörg Ott	jo@tzi.org
Carsten Bormann	cabo@tzi.org

Overview

- Recap SDPng Model
  - as of draft-kutscher-mmusic-sdpng-req-01.txt
- Strawman proposal for SDPng syntax
- Next steps
- Discussion

Proposed Architecture for Description Language

Four different sections



Definitions Section

- Contains definitions of abstractions for later re-use
  - For example codecs, redundancy schemes, transport mechanisms, payload formats
- Can be factored-out
  - Profiles
    - Referenced by description documents
    - AVP registry like, for well-known basic definitions
    - But applications should be allowed to reference arbitrary external profile definitions
- Include by reference or “inline”

Potential and Actual Configurations

- Definition of possible configurations (capabilities)
- Combine basic definitions to create potential configurations for different components
  - Reuse definitions from first section
  - Label different configurations with unique identifiers for later referencing

Constraints

- Specifies constraints for combination of configurations
  - E.G.: When doing X, I cannot only support Y but not Z at the same time
- Why separate from configuration section?
  - Constraints can apply across *combinations* of configurations
  - Backward “compatibility” (mapping to SDP)
  - Simplifies minimal implementations
- Section can be empty…

Session Attributes

- SDP-like “session level” parameters.
  - Title, date, duration, contact info etc.
- In general:
  - Meta-information about the conference *and* the media streams.
  - “Stream A is speaker-video, stream B is slide presentation.”
    - SDP requires implicit assumptions.

Related Work

- CC/PP
  - Functionality
    - Framework for expressing capabilities and preferences of clients (user agents)
    - Origin servers selects/produces appropriate content according to supplied user agent profile in requests
  - Implementation
    - Use of RDF/XML to express capabilities/preferences
- SMIL
  - Synchronized Multimedia Integration Language
  - Modular specification (a set of XML-DTDs)
  - Presentation descriptions are XML documents

SDPng Syntax

- XML
  - Structured description instances
  - Formal validation (of specifications)
    - DTDs, schemas provide rules
  - Extensibility
    - Namespace concept
  - Ubiquity
    - Parsers may already be implemented in end systems

SDPng Model in XML

Definitions	Elements for defining codecs, transport mechanisms, referring to existing definitions
Potential and actual Configurations	Elements for listing configurations that are combinations of definitions (may either reference or define inline)
Constraints	Reference configurations and express constraints on combinations, number of instantiations etc.
Session Attributes	Elements for meta information on individual applications (i.e., streams, sessions) that reference configuration definitions.

Two Types of External “Packages”

- Rules Packages
  - Defining extensions for certain application
    - “Transport mechanism X requires these parameters…”
- Definition Packages
  - Collections of concrete definitions
  - Media Formats
    - E.g., commonly used audio formats
  - Transport Mechanisms
    - RTP/UDP/IP with AVP
- Issue:
  - Registry for commonly used rules and definitions?

External Definitions

- Uniqueness of names
  - Avoiding name clashes requires namespace concept
- Referencing external packages in SDPng instances
  - Implementations MUST be able to retrieve external definitions?
  - Or: Create a registry for common definitions and require extensions always to be provided inline?

Definition Mechanisms for XML

- XML Document Type Definitions
  - Provide definition mechanisms for element content and attribute types
- XML Schemas
  - Allow more fine-grained control of content model and attribute values
  - Provides concept of modularity and extensibility of schema definitions
    - Schema definitions can be
      - Combined with other schema definitions
      - Extend or restrict other schema definitions
    - Relies on XML namespace mechanism

Example: Definitions

Reusing definitions by referencing them

```
<audio-codec name="audio-basic" encoding="PCMU“
 sampling_rate="8000” channels="1"/>

<audio-codec name="audio-L16-mono" encoding="L16“
 sampling_rate="44100” channels="1"/>

<fec name="parityfec“/>


<audio-red name="red-pcm-gsm-fec">
  <use ref="audio-basic"/>
  <use ref="audio-gsm"/>
  <use ref="parityfec"/>
</audio-red>
```

Example: Configurations

```
<cfg>
  <component name="audio1" media="audio">
    <alt name="AVP-audio-0">
      <rtp transport="udp-ip" format="audio-basic">
        <addr type="mc">
          <ipv4>239.239.239.239</ipv4> <port>30000</port>
        </addr>
      </rtp>
    </alt>

    <alt name="AVP-audio-11">
      <rtp transport="udp-ip" format="audio-L16-mono">
        <addr type="mc">
          <ipv4>239.239.239.239</ipv4> <port>30000</port>
        </addr>
      </rtp>
    </alt>
  </component>
</cfg>
```

Example: Constraints

Proposal:

- Limit functionality to basic constraints
- Implementations SHOULD consider constraints…

```
<constraints>
  <par>
    <use ref="AVP-audio-11" max="5">
    <use ref="AVP-video-32" max="1">
  </par>
</constraints>
```

Example: Meta-Information

- Re-use of existing meta information efforts?
  - SMIL-2.0 advocates use of RDF…

```
<conf>
  <subject>SDPng test</subject>
  <originator>joe@example.com</originator>
  <about>A test conference</about>
  <info name="audio1" function="speaker">
    Video stream for the different speakers…
  <info>
</conf>
```

Summary

- SDPng uses XML
  - Structured information
  - Namespace concept
  - Definition mechanisms (DTDs, Schema)

- Next steps
  - Work on definition mechanism
  - Define packages for common use cases
  - Capability negotiation
  - draft-ietf-mmusic-sdpng-00.txt ASAP