


Common Announcement Interchange Format

Oliver Goebel Anselm R. Garbe

RUS-CERT University of Stuttgart

17th July 2004

RUS  CERT



CAIF Changes

- 1 CAIF Changes
 - Changes from 1.0 to 1.1
 - Planned changes
 - Discussion

Changes from 1.0 to 1.1

- multi-targetgroup: **target-groups**
- multi-linguality: **body**
- normalization of *attack-information*
- new element: **constituency**
- new element: **interchange**
- new element: **earliest-release**

target-groups (DTD)

```
<!ELEMENT target-groups (target-group+)  
>
```

```
<!ELEMENT target-group          %MTEXT;>
```

```
<!ATTLIST target-group
```

id	ID	#REQUIRED
lang	%LANG_CODE;	#REQUIRED
tech-background	%TECH-BACKGROUND;	"admin"
orga-overview	%ORGA-OVERVIEW;	#IMPLIED
environment	%ATEXT;	#IMPLIED
headline	%ATEXT;	#IMPLIED>

ORGA-OVERVIEW = [employee | middle-management | senior-management]

TECH-BACKGROUND = [admin | user]

target-groups (example)

```
<target-groups>
  <target-group id="en-admin"
    lang="en-us"
    tech-background="admin"
    headline="Who should read this document?">
    This document should be read by Microsoft
    Windows system administrators.
  </target-group>
  <target-group id="de-admin"
    lang="de"
    tech-background="admin">
    ...
  </target-group>
</target-groups>
```

new element: body

DTD:

```
<!ELEMENT body %MTEXT;>
<!ATTLIST body
    tg-id          IDREFS          #IMPLIED
    headline       %ATEXT;        #IMPLIED
>
```

Example:

```
<body tg-ref="en-admin" headline="Revisions of this Announcement"/>
<body tg-ref="de-admin" headline="Revisionen dieser Meldung"/>
```

attack-information normalization

Old (DTD):

```
<!ELEMENT attack-information (  
    headline?,  
    vector,  
    requirements?,  
    signature?  
)  
>
```

attack-information normalization

New (DTD):

```
<!ELEMENT problem-id  
    (  
        body+,  
        class?,  
        attack-vector?,  
        attack-requirements?,  
        attack-signature?,  
        impact?,  
        exploit-status?,  
        risk?,  
        probability-of-occurrence*,  
        threat*  
    )>
```


new element: constituency (DTD)

Defining element:

```
<!ELEMENT constituencies (constituency+)>
  <!ELEMENT constituency (body+)>
  <!ATTLIST constituency
    id ID #REQUIRED
  >
```

Markup element:

```
<!ELEMENT const %MTEXT;>
<!ATTLIST const
  ref IDREFS #REQUIRED
  >
```

new element: constituency (example)

```
<constituencies>
  <constituency id="Uni-Stuttgart">
    <body tg-ref="en-admin">
      Stuttgart University
    </body>
    <body tg-ref="de-admin">
      Universität Stuttgart.
    </body>
  </constituency>
  ...
</constituencies>
...
<body tg-ref="de-admin">
  <const ref="Uni-Stuttgart">
    Dieses Problem betrifft unsere studentischen Service-Systeme.
  </const>
</body>
```

new element: interchange

DTD:

```
<!ELEMENT interchange EMPTY>
<!ATTLIST interchange
    restriction    %RESTRICTION-KEYS;  "none"
>
```

Example:

```
<identification>
  ...
  <interchange restriction="constituency"/>
</identification>
```

RESTRICTION-KEYS = [none | constituency]

new element: earliest-release

DTD:

```
<!ELEMENT earliest-release    (%UTEXT;)>
<!ATTLIST earliest-release
    date                %DATE;          #REQUIRED
>
```

Example:

```
<identification>
  ...
  <earliest-release date="2004-07-26">
    Initial release.
  </earliest-release>
</identification>
```

Planned changes

- new element: **aff-definition**
- new element: **determine-affectedness**

Planned changes cont'd

Defining element:

```
<!ELEMENT determine-affectedness    %MTEXT>
<!ATTLIST determine-affectedness
    method          %METHOD-KEYS;    "script"
    aff-ref          IDREFS          #REQUIRED
>
```

Markup element:

```
<!ELEMENT aff-definition            %MTEXT;>
<!ATTLIST aff-definition
    id              ID              #REQUIRED
>
```

METHOD-KEYS = [script | shell | ...]

Discussion

Questions?

CAIF Authoring Tool

- 2 Requirements
 - Hardware Requirements
 - Software Requirements
 - UseCases
- 3 Design
 - Architecture
 - Interfaces
- 4 Implementation
 - State-of-the-art
 - Roadmap
 - Demo
 - Discussion
- 5 Further Reading

Hardware Requirements

- PC or Workstation with **256 MB RAM**
- **1 GB** disk space for database storage
- Permanent network connection

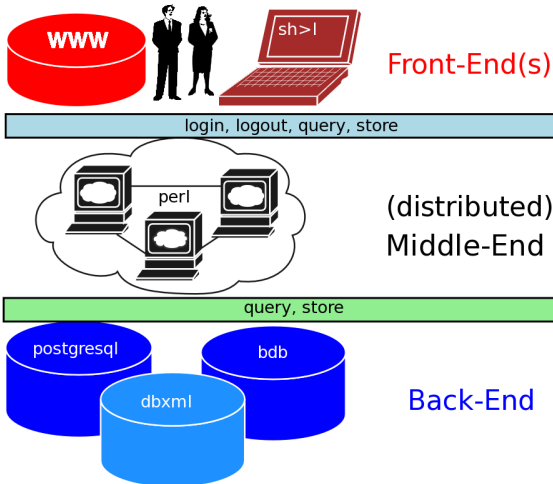
Software Requirements

- POSIX compatible operating system (Linux or UNIX-alike)
- PostgreSQL database
- Berkeley DB (for session management)
- Sleepycat XML DB (optional)
- Apache 1.3.x with mod_perl (Apache 2.0 planned)
- Perl 5.x installation with DBI and XML packages
- AxKit for mod_perl
- libxml for various XML/XSLT processing

UseCases

- UC *Search/View* for announcements
- UC *Export* an announcement
- UC *Login/Logout* for write access
- UC *Create* new/existing announcement
- UC *Review* an announcement
- UC *Mark* an announcement as **deprecated**, **(un)released**, **reviewed**

3-Tier Architecture



Front-End - Middle-End Interface

- login:
usercontent login(username, password)
- logout:
undefined logout(usercontext)
- query (without user context):
resultset query(pattern)
- query (with user context):
resultset query(usercontext, pattern)
- store (ACID¹):
boolean store(usercontext, content/states)

Front-End - Middle-End Interface

- login:
usercontent login(username, password)
- logout:
undefined logout(usercontext)
- query (without user context):
resultset query(pattern)
- query (with user context):
resultset query(usercontext, pattern)
- store (ACID¹):
boolean store(usercontext, content/states)

Front-End - Middle-End Interface

- login:
usercontent login(username, password)
- logout:
undefined logout(usercontext)
- query (without user context):
resultset query(pattern)
- query (with user context):
resultset query(usercontext, pattern)
- store (ACID¹):
boolean store(usercontext, content/states)

¹Atomicity, Consistency, Isolation, Durability

Front-End - Middle-End Interface

- login:
usercontent login(username, password)
- logout:
undefined logout(usercontext)
- query (without user context):
resultset query(pattern)
- query (with user context):
resultset query(usercontext, pattern)
- store (ACID¹):
boolean store(usercontext, content/states)

¹Atomicity, Consistency, Isolation, Durability

Front-End - Middle-End Interface

- login:
usercontent login(username, password)
- logout:
undefined logout(usercontext)
- query (without user context):
resultset query(pattern)
- query (with user context):
resultset query(usercontext, pattern)
- store (ACID¹):
boolean store(usercontext, content/states)

¹Atomicity, Consistency, Isolation, Durability

Middle-End - Back-End Interface

- query (with user context):
resultset query(user context, pattern)
- store:
boolean store(usercontext, content/states)

Middle-End - Back-End Interface

- query (with user context):
resultset query(user context, pattern)
- store:
boolean store(usercontext, content/states)

State-of-the-art

- Front-End: rudimentary XSLT processing implemented (AxKit based) **40%**
- Middle-End: currently no session management, only delegation **10%**
- Back-End: CAIF based RDB-XML layer **90%**,
SleepyCat DBXML experimental

State-of-the-art cont'd

```
src
|----CAIF
|----|----DB
|----|----|----Middleware
|----|----|----|----RDB.pm
|----|----|----Manager.pm
|----|----Tools
|----|----|----sql2xmldbmsmap.pl
|----|----XML
|----|----|----XSLT
|----|----|----stylesheets
|----|----|----|----caif.xsl
|----|----Web
|----|----|----sample.xml
|----|----|----README
|----|----|----caif.xsl
|----|----|----WebManager.pl
|----|----Middleware
|----|----|----Delegator.pm
```

Front-End State-of-the-art

- mod_perl und AxKit
- XSLT of CAIF documents/fragments
- XML-RPC for queries/stores

Middle-End State-of-the-art

- Perl server process
- *communication*: XML-RPC
- Validation of all data
- Session Management (Berkeley DB)

Back-End State-of-the-art

- RDB-XML Transformation
- *optional*: Sleepycat DBXML

Roadmap

- Begin of August: basic prototype (single user), first public snapshot **alpha**
- Mid of September: extended prototype (multi-user middle-end, session management) **beta**
- End of October: first public release (synchronization mechanisms between distributed middle-ends) **production**

Roadmap

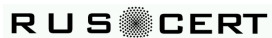
- Begin of August: basic prototype (single user), first public snapshot **alpha**
- Mid of September: extended prototype (multi-user middle-end, session management) **beta**
- End of October: first public release (synchronization mechanisms between distributed middle-ends) **production**

Roadmap

- Begin of August: basic prototype (single user), first public snapshot **alpha**
- Mid of September: extended prototype (multi-user middle-end, session management) **beta**
- End of October: first public release (synchronization mechanisms between distributed middle-ends) **production**

Short demo

Short demo, browsing the code...



Discussion

Questions?

For Further Reading

-  Oliver Goebel, Florian Weimer
CAIF Requirements
-  Oliver Goebel
CAIF Format Specification
-  Anselm R. Garbe
CAIF Authoring Tool Specification
-  Anselm R. Garbe
CAIF Authoring Tool Design