Automatic upconversion using XSLT 2.0 and XProc
A real world example

Stefanie Haupt
Maik Stührenberg
Bielefeld University

Overview

- Introduction
  - A perfect (?) holiday season
  - Data on the web
- Our corpus of video game reviews
- Upconversion using XSLT 2.0 and XProc
  - Creating an XSD for game reviews
  - Processing the corpus
- Querying
- Future prospects

Introduction

- Kids love video games these days ...
  and often they leave their parents far behind
- Choosing the right video game as holiday present is not an easy task...

Source:http://cdn.physorg.com/newman/gfx/news/hires/1-nintendoprof.jpg

Rayman © Ubisoft

Introduction

... which can turn out horribly wrong

by buying the game according solely to the information given on the packaging

Source: http://kotaku.com/196594/psp-makes-children-cry (modified)

Introduction

... which can turn out horribly wrong

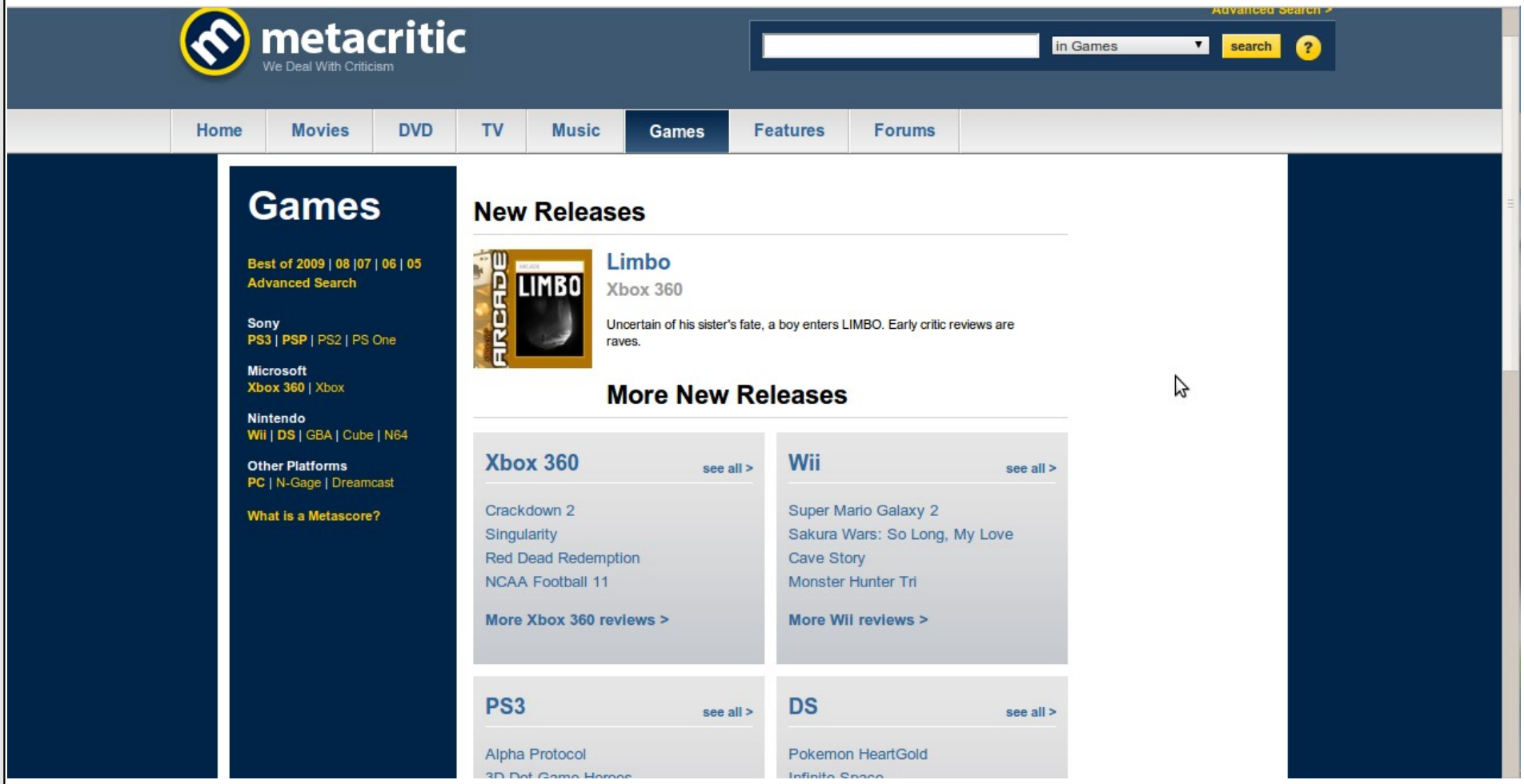
by buying the game according solely to the information given on the packaging

Source: http://kotaku.com/196594/psp-makes-children-cry (modified)

Introduction

Nowadays there's information everywhere on the web

As are a lot of video game reviews...

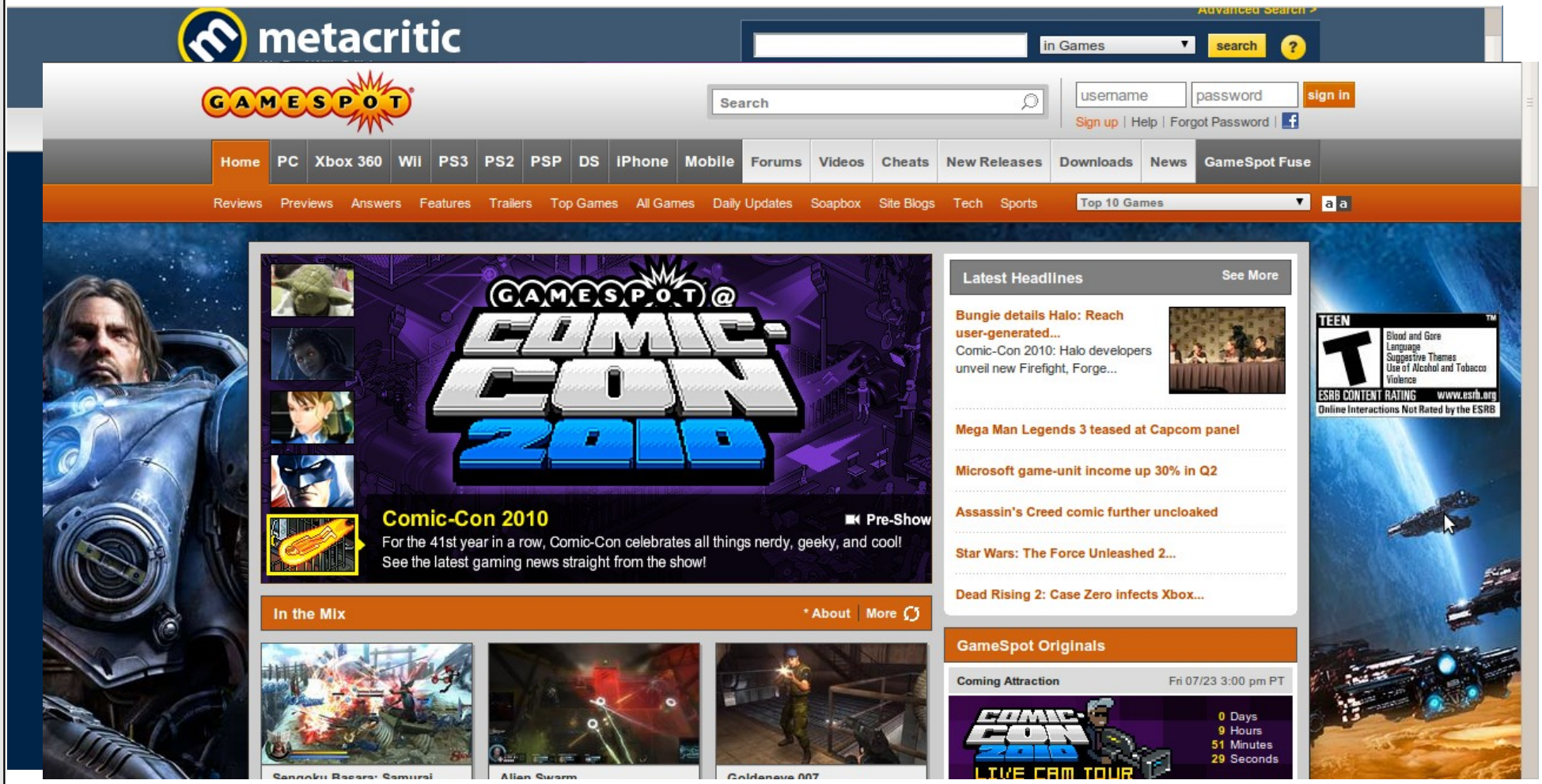


Source: http://www.metacritic.com/games

Introduction

Nowadays there's information everywhere on the web

As are a lot of video game reviews...



Source: http://www.gamespot.com/

Introduction

Nowadays there's information everywhere on the web

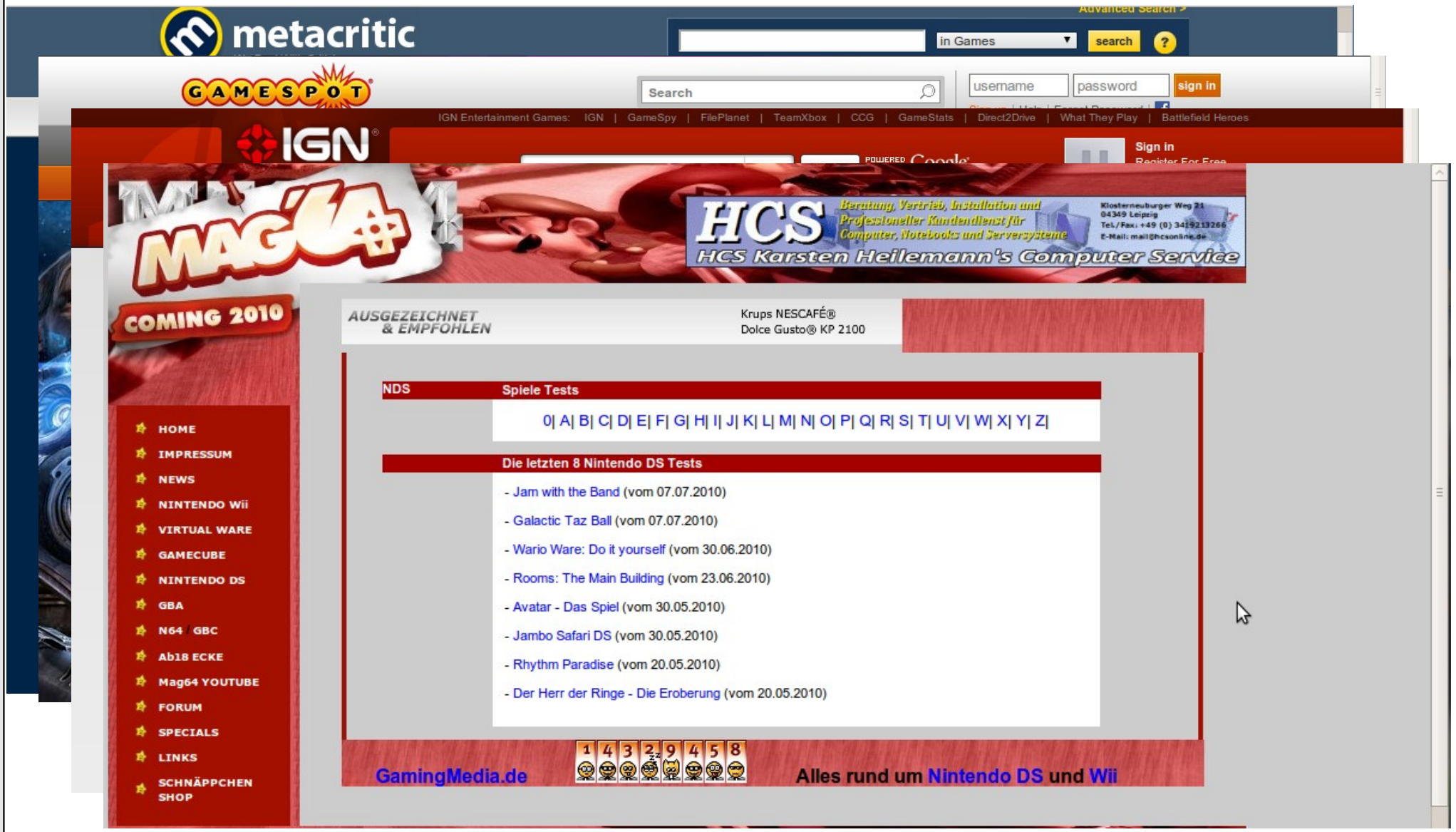
As are a lot of video game reviews...

Source: http://uk.ign.com/

Introduction

Nowadays there's information everywhere on the web

As are a lot of video game reviews...



Source: http://mag64.de/

Introduction – Data on the web

- As millions of other sites the German Mag'64 team publishes its content – mainly video game reviews – in HTML
- As a hypertext language HTML does not offer a large pool of semantically motivated elements or attributes for annotating arbitrary data
- Although CSS microformats can be used, most information is buried beneath a "tag soup" of `<td>`, `<p>` and `<div>` elements
- Our goal is to make very specific information easy to retrieve AND stay in the XML realm

Our corpus of video game reviews

- The German Mag'64 team publishes video game reviews for Nintendo consoles for several years
- Most of these reviews are written in invalid HTML 4.01 with often wrong declared character set
- There are two types of reviews which differ not much regarding the content - but the markup
  - Type A (~2001 - 2004)
  - Type B (since 2004)
- Taken together there are more than 1500 reviews with valuable but hard to find information

Our corpus of video game reviews

But which information is valuable?

- Title of the game
- Hardware requirements
- Genre and number of players
- Age rating
- Difficulty
- ...
- Is it Fun?

Our corpus of video game reviews

This is what a typical Type A review looks like

SYSTEM:	*GCN - PAL*	SCHWIERIGKEIT:	*1-6*
ENTWICKLER:	*Ubi Soft*	SECRETS:	*JA*
GENRE:	*Jump'n Run*	SPRACHHÜRDE:	*Keine*
SPIELER:	*1-4 Spieler*	PREIS:	*ca.60 Euro*
HANDBUCH:	*Mehrsprachig*	TERMIN:	*Erhältlich*
MEMCARD:	*8 Seiten*		
60Hz Modus:	*JA*		

Dieser Testbericht wurde geschrieben von *Matthias Engert* am 24.02.2003

CHEATS: JA	KOMPLETTLÖSUNG: JA	TIME TRIAL / SCORES: JA

Bisher hat uns Ubi Soft ja zumeist mit Spielen anderer Firmen mehr oder weniger beglückt. Eigenentwicklungen waren noch recht wenige darunter. Dabei hat Ubi Soft doch einiges zu bieten. Unter anderem niemand geringeren als Rayman. Einen Charakter den wohl die meisten unter euch kennen und spielerisch schonmal begutachtet haben. Zumindest waren die Rayman Spiele immer recht erfolgreich.

Source: http://www.n2002.de/ngc/rayman3/rayman3.htm

Our corpus of video game reviews

This is what a typical Type A review looks like

SYSTEM:	*GCN - PAL*	SCHWIERIGKEIT:	*1-6*
ENTWICKLER:	*Ubi Soft*	SECRETS:	*JA*
GENRE:	*Jump'n Run*	SPRACHHÜRDE:	*Keine*
SPIELER:	*1-4 Spieler*	PREIS:	*ca.60 Euro*
HANDBUCH:	*Mehrsprachig*	TERMIN:	*Erhältlich*
MEMCARD:	*8 Seiten*		
60Hz Modus:	*JA*		

Dieser Testbericht wurde geschrieben von *Matthias Engert* am 24.02.2003

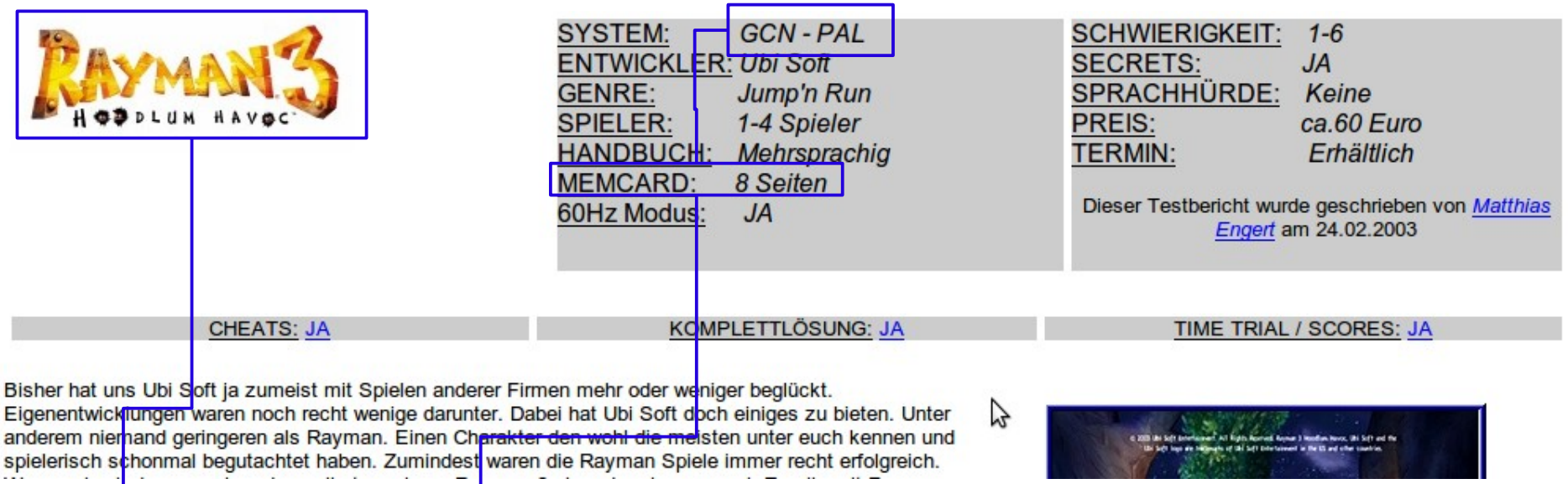
CHEATS: JA | KOMPLETTLÖSUNG: JA | TIME TRIAL / SCORES: JA

Bisher hat uns Ubi Soft ja zumeist mit Spielen anderer Firmen mehr oder weniger beglückt. Eigenentwicklungen waren noch recht wenige darunter. Dabei hat Ubi Soft doch einiges zu bieten. Unter anderem niemand geringeren als Rayman. Einen Charakter den wohl die meisten unter euch kennen und spielerisch schonmal begutachtet haben. Zumindest waren die Rayman Spiele immer recht erfolgreich.

Title

Our corpus of video game reviews

This is what a typical Type A review looks like

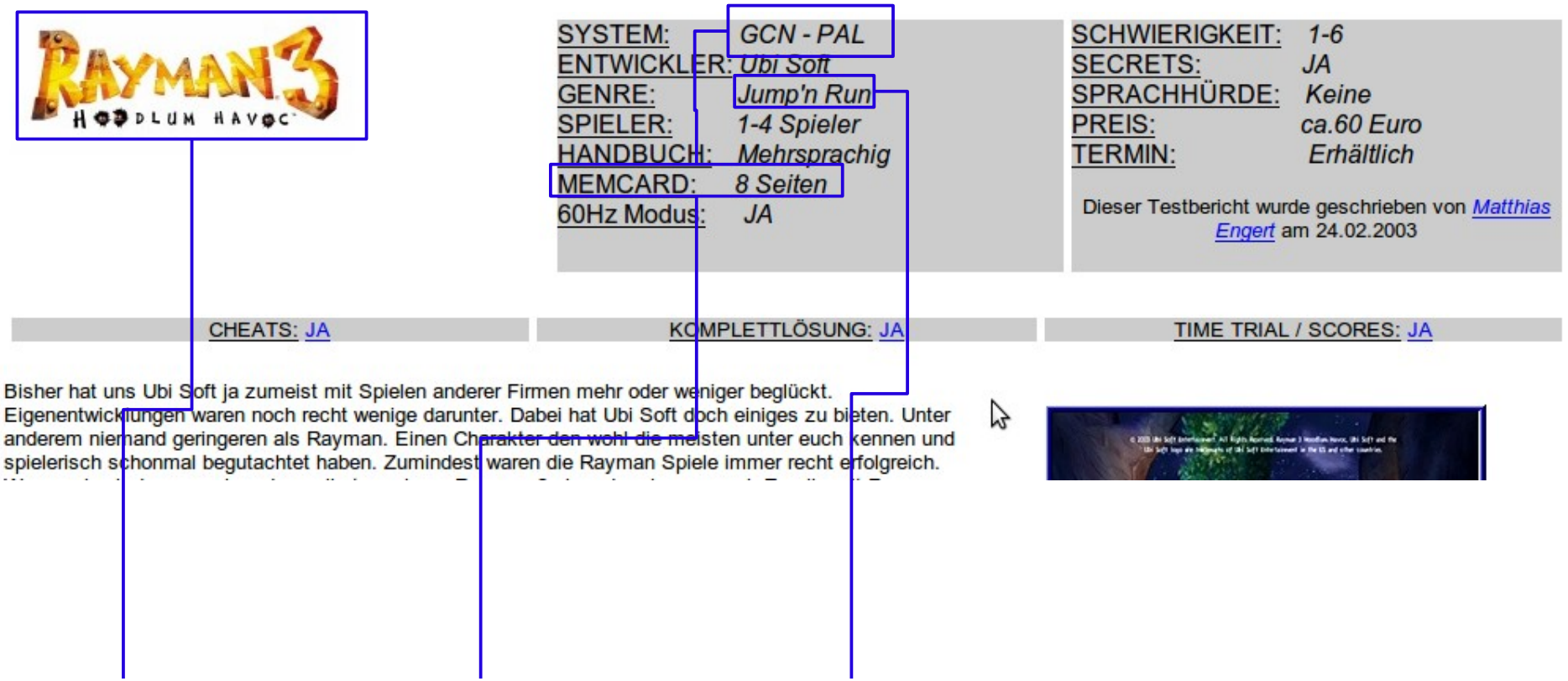


Title

Hardware requirements

Our corpus of video game reviews

This is what a typical Type A review looks like



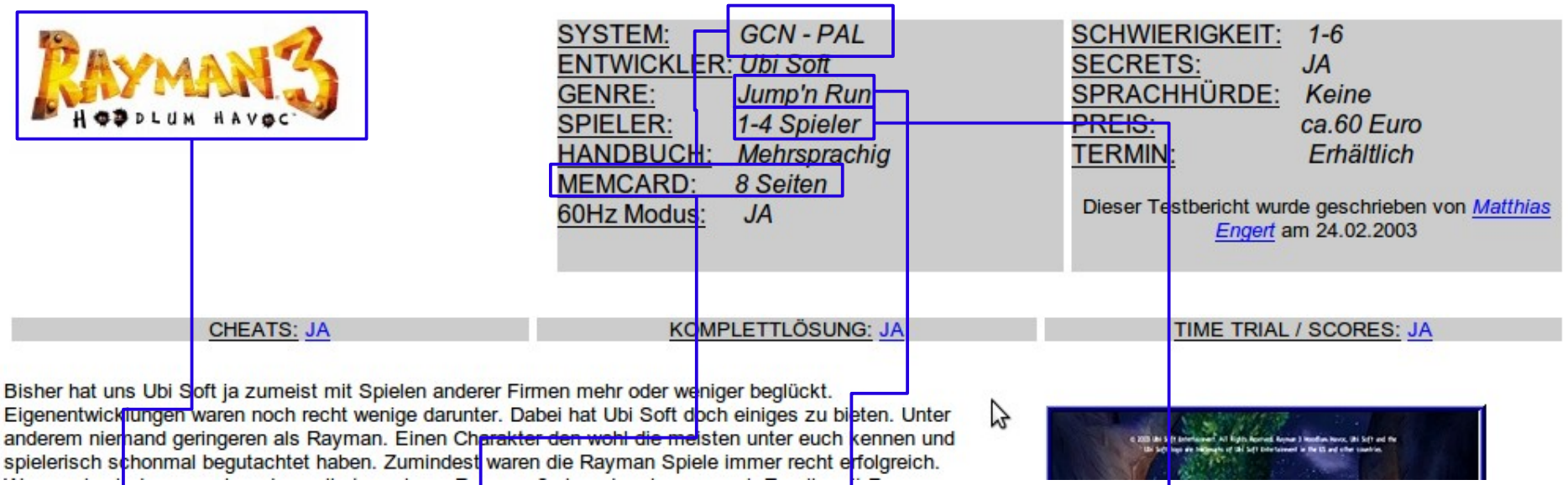
Title

Hardware requirements

Genre

Our corpus of video game reviews

This is what a typical Type A review looks like



Title

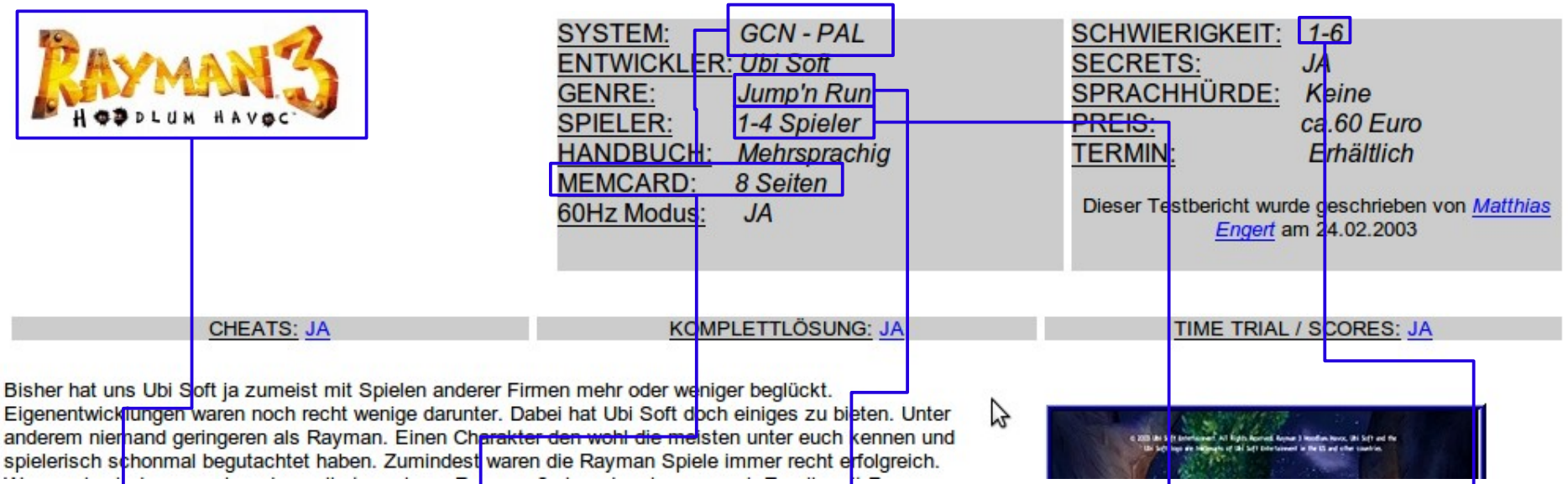
Hardware requirements

Genre

Number of players

Our corpus of video game reviews

This is what a typical Type A review looks like



Title

Hardware requirements

Genre

Number of players

Difficulty

A closer look at a Type A review

- The markup is very presentation focused and often even this information is doubled

```
<tr>
 <td width="35%" align="left" valign="top">
  <img src="ray3logo.jpg">
 </td>
 <td width="33%" align="left" valign="top" bgcolor="#CCCCCC">
  <p>
   <font face="Arial, Helvetica, sans-serif" size="3">
    <u>
     <font face="Arial, Helvetica, sans-serif" size="3">SYSTEM:</font>
    </u>
    <font face="Arial, Helvetica, sans-serif" size="3">
             
       
     <i>GCN - PAL</i>
    </font>
    <u><br>ENTWICKLER:</u>
    <i>Ubi Soft</i>
   </font><br>
  <u>
  <font face="Arial, Helvetica, sans-serif" size="3">GENRE:</font>
  </u>
<!-- [...] -->
```

Source: http://www.n2002.de/ngc/rayman3/rayman3.htm

A closer look at a Type A review

- The markup is very presentation focused and often even this information is doubled
- The title of the reviewed game is represented by an image, the tag lacking the required alt-attribute, apart from this, it is hidden inside the running text without markup of any kind

```
<tr>
 <td width="35%" align="left" valign="top">
  <img src="ray3logo.jpg">
 </td>
<!-- [...] -->
</tr>
<!-- [...] -->
<table width="98%" border="0" cellspacing="0" cellpadding="0" align="center">
<!-- [...] -->
 <td width="60%" align="left" valign="top">
<font face="Arial, Helvetica, sans-serif" size="2">
<!-- [...] -->
  Story verfolgen. Rayman 3 Hoodlum Havoc beginnt damit,
<!-- [...] -->
```

Source: http://www.n2002.de/ngc/rayman3/rayman3.htm

A short look at a Type B review

- The main differences to a Type A review are
  - Title of the game in prominent textual display
  - Information about age rating provided
  - The running text is subdivided by headings
  - An external CSS stylesheet is used

Source: http://mag64.de/index.php?page=nds/cnr/cnr&server=3&ext=1

A short look at a Type B review

- The main differences to a Type A review are
  - Title of the game in prominent display
  - Information about age rating provided
  - The running text is subdivided by headings
  - An external CSS stylesheet is used

A short look at a Type B review

- The main differences to a Type A review are
  - Title of the game in prominent display
  - Information about age rating provided
  - The running text is subdivided by headings
  - An external CSS stylesheet is used


Cartoon Network Racing - NDS

Matthias Engert (11.03.2007)

SYSTEM: NDS
ENTWICKLER: firebrand
GENRE: Funracer
SPIELER: 1-4 Spieler
HANDBUCH: Mehrsprachig
SPEICHER: Batterie
1MODUL MP: Ja
SCHWIERIGKEIT: 1-7
SECRETS: Ja

SPRACHHÜRDE: Keine
MIKRO SUPPORT: Nein
ALTERSFREIGABE: 3+
TERMIN: Erhältlich
VIRTUAL SURROUND: Nein
PREIS: ca.40 Euro
KOMPLETTLÖSUNG: Nein
CHEATS / TIPPS: Nein
LESERMEINUNGEN: Nein

Einleitung....

Im Bereich der Funracer haben die Handheld Konsolen den TV Brüdern

A short look at a Type B review

- The main differences to a Type A review are
  - Title of the game in prominent display
  - Information about age rating provided
  - The running text is subdivided by headings
  - An external CSS stylesheet is used but still a lot of local design markup is used anyway

```
<head>
 <title>NDS Cartoon Network Racing</title>
 <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
 <link rel="stylesheet" href="http://www.mag64.de/test.css" type="text/css">
</head>
<body marginwidth="0" marginheight="0" leftmargin="0" topmargin="0"
bgcolor="#CCCCCC">
<table width="710" border="0" cellpadding="0" cellspacing="0" bgcolor="#CCCCCC">
```

In a perfect world

- Information would be structured well
- And easy to find

Upconversion

Let's equip our power gloves:

- Creating an XSD for game reviews
- Processing the corpus

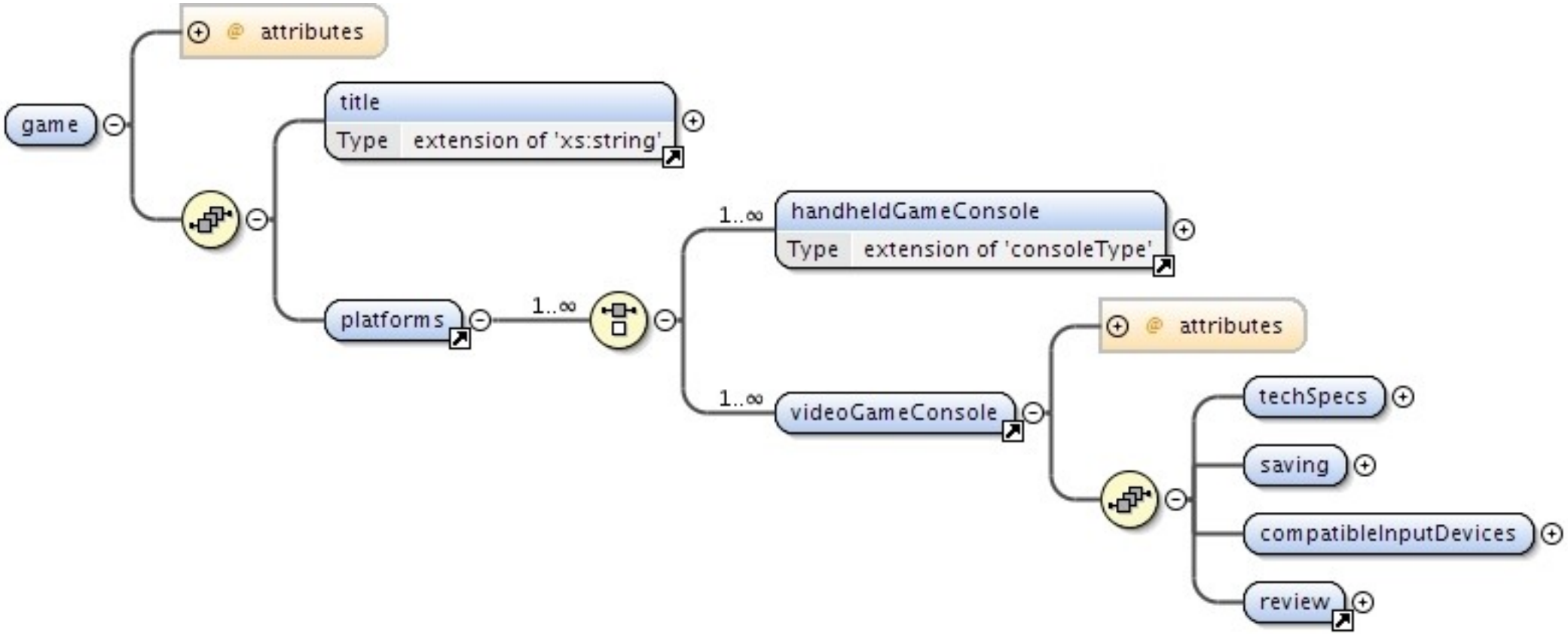
Creating an XSD for game reviews

Why XML schema and not DTD or RELAX NG?

- Datatype library
- Support of user-defined simple and complex types
- Broad processor support

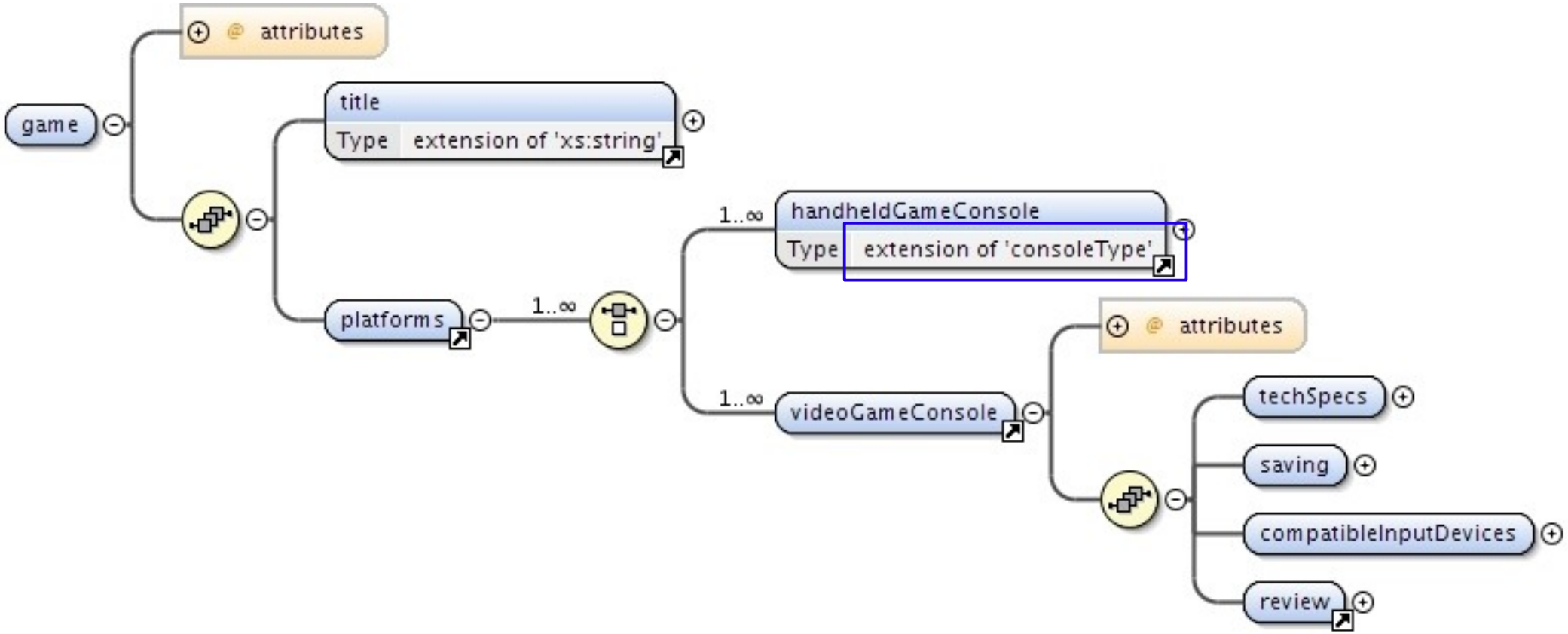
Creating an XSD for game reviews

- Focus on single game
- Each game can be identified by a unique `xml:id` attribute
- Optional attributes correspond to `genre` and `subgenre`, supporting an enumerated list of possible values
- Children of the `game` element are the `title` and `platforms` elements



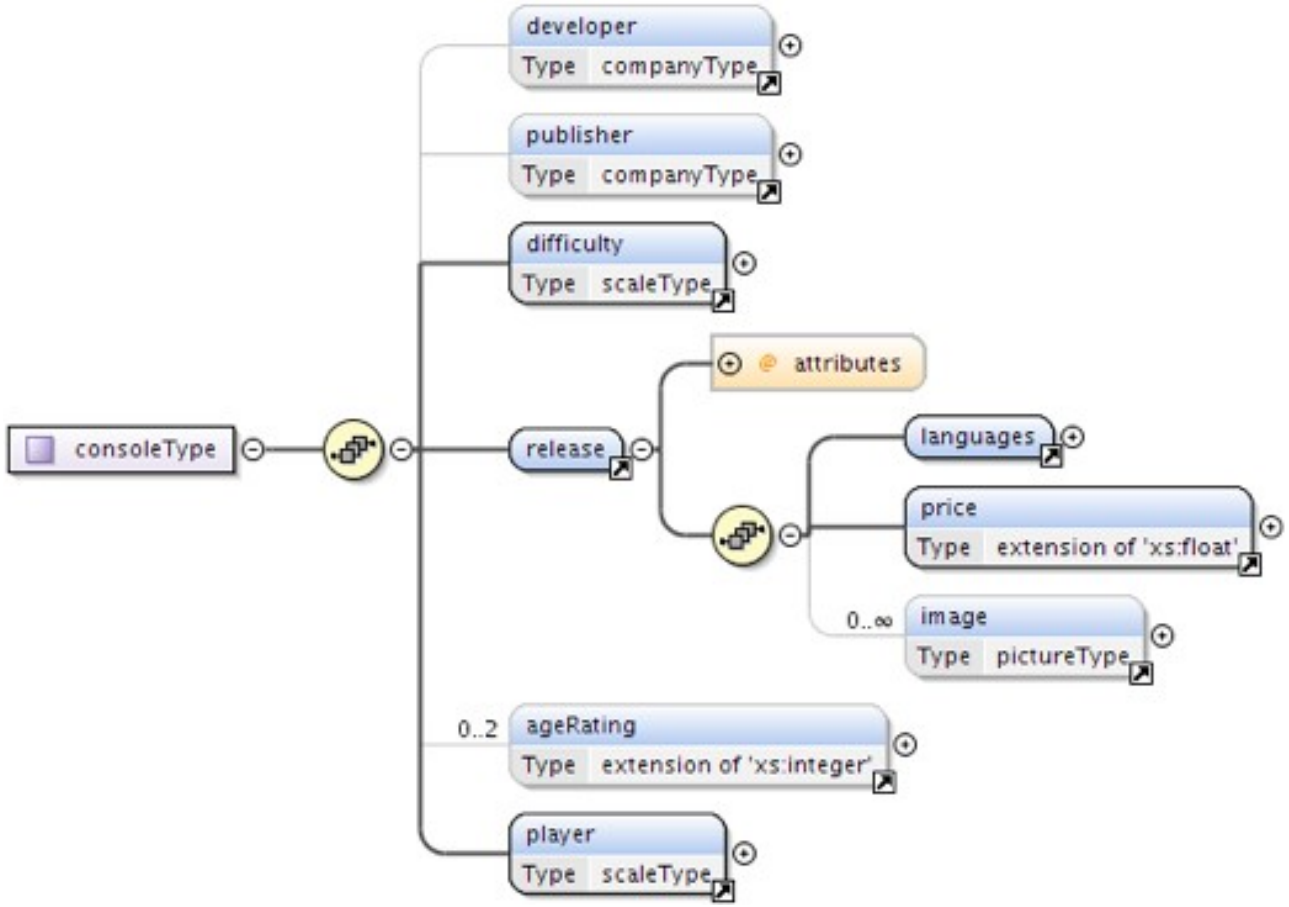
Creating an XSD for game reviews

- Focus on single game
- Each game can be identified by a unique `xml:id` attribute
- Optional attributes correspond to `genre` and `subgenre`, supporting an enumerated list of possible values
- Children of the `game` element are the `title` and `platforms` elements



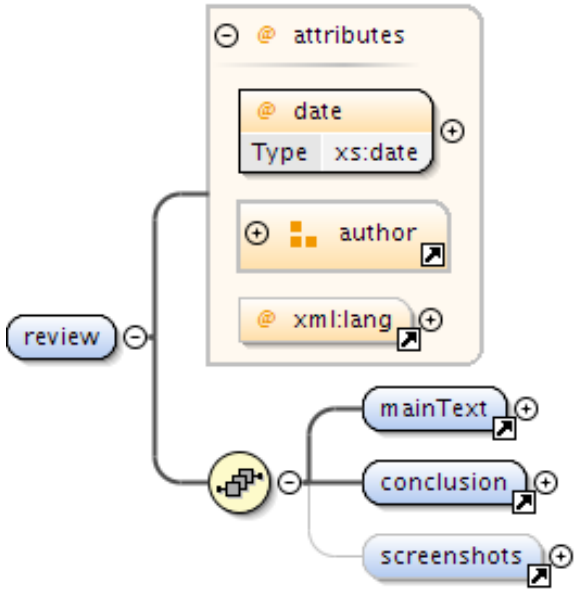
Creating an XSD for game reviews - the consoleType

- The complex type `consoleType` provides the basis for the elements `handheldGameConsole` and `videoGameConsole`
- We extend it with special features for handheld and stationary consoles respectively



Creating an XSD for game reviews - the review element

- `mainText` holds the running text consisting of optional headers and paragraphs
- `conclusion` stores both further text (e.g. in a form of a final verdict) and the tabular-like lists of pros and cons, followed by the final `score` element
- Scoring can be expressed
  - either via numeric values using the `percent` child element with its attributes `graphics`, `sound` and `multiplayer` (both optional) and `overall`
  - or by text



Creating an XSD for game reviews

To sum up:

- Our XSD fits the needs of both Type A and Type B reviews
- It provides various enumerated lists to avoid typos and misspellings
- It allows to combine reviews of the same video game released on different platforms
- It is extendable and flexible

Let's take a look at a possible result

```
<?xml version="1.0" encoding="UTF-8"?>
<game xml:id="d1" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:noNamespaceSchemaLocation="Struktur.xsd" genre="Jump 'n' Run">
 <title abbreviation="rayman3">Rayman3 Hoodlum Havoc</title>
 <platforms>
  <videoGameConsole type="GCN">
   <developer>Ubi Soft</developer>
   <difficulty min="1" max="6"/>
   <release>
    <languages>
     <spoken xml:lang="de"/>
    </languages>
    <price currency="EUR">60</price>
   </release>
   <player min="1" max="4"/>
   <techSpecs> <item>PAL</item> </techSpecs>
   <saving mode="Memorycard" blocks="8"/>
   <compatibleInputDevices>
    <item>Gamecube Controller</item>
    <item>GBA</item>
   </compatibleInputDevices>
   <review date="2003-02-24" authorFirstname="Matthias" authorLastname="Engert">
    <mainText>
     <paragraph>Bisher hat uns Ubi Soft ja (...)</paragraph>
    </mainText>
    <conclusion>
     <pro>
      <item>Unterhaltsames Gameplay</item>
     </pro>
     <contra>
      <item>Ende wird zu schnell erreicht</item>
     </contra>
     <score>
      <percent graphics="85" sound="85" multiplayer="82" overall="82"/>
     </score>
    </conclusion>
   </review>
  </videoGameConsole>
 </platforms>
</game>
```

Let's take a look at a possible result

```
<?xml version="1.0" encoding="UTF-8"?>
<game xml:id="d1" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:noNamespaceSchemaLocation="Struktur.xsd" genre="Jump 'n' Run">
 <title abbreviation="rayman3">Rayman3 Hoodlum Havoc</title>
 <platforms>
  <videoGameConsole type="GCN">
   <developer>Ubi Soft</developer>
   <difficulty min="1" max="6"/>
   <release>
    <languages>
     <spoken xml:lang="de"/>
    </languages>
    <price currency="EUR">60</price>
   </release>
   <player min="1" max="4"/>
   <techSpecs> <item>PAL</item> </techSpecs>
   <saving mode="Memorycard" blocks="8"/>
   <compatibleInputDevices>
    <item>Gamecube Controller</item>
    <item>GBA</item>
   </compatibleInputDevices>
   <review date="2003-02-24" authorFirstname="Matthias" authorLastname="Engert">
    <mainText>
     <paragraph>Bisher hat uns Ubi Soft ja (...)</paragraph>
    </mainText>
    <conclusion>
     <pro>
      <item>Unterhaltsames Gameplay</item>
     </pro>
     <contra>
      <item>Ende wird zu schnell erreicht</item>
     </contra>
     <score>
      <percent graphics="85" sound="85" multiplayer="82" overall="82"/>
     </score>
    </conclusion>
   </review>
  </videoGameConsole>
 </platforms>
</game>
```

Let's take a look at a possible result

```
<?xml version="1.0" encoding="UTF-8"?>
<game xml:id="d1" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:noNamespaceSchemaLocation="Struktur.xsd" genre="Jump 'n' Run">
 <title abbreviation="rayman3">Rayman3 Hoodlum Havoc</title>
 <platforms>
  <videoGameConsole type="GCN">
   <developer>Ubi Soft</developer>
   <difficulty min="1" max="6"/>
   <release>
    <languages>
     <spoken xml:lang="de"/>
    </languages>
    <price currency="EUR">60</price>
   </release>
   <player min="1" max="4"/>
   <techSpecs> <item>PAL</item> </techSpecs>
   <saving mode="Memorycard" blocks="8"/>
   <compatibleInputDevices>
    <item>Gamecube Controller</item>
    <item>GBA</item>
   </compatibleInputDevices>
   <review date="2003-02-24" authorFirstname="Matthias" authorLastname="Engert">
    <mainText>
     <paragraph>Bisher hat uns Ubi Soft ja (...)</paragraph>
    </mainText>
    <conclusion>
     <pro>
      <item>Unterhaltsames Gameplay</item>
     </pro>
     <contra>
      <item>Ende wird zu schnell erreicht</item>
     </contra>
     <score>
      <percent graphics="85" sound="85" multiplayer="82" overall="82"/>
     </score>
    </conclusion>
   </review>
  </videoGameConsole>
 </platforms>
</game>
```

Processing the corpus - XSLT 2.0

We process the corpus by using XSLT 2.0 / XPath 2.0

Key features which produce benefit:

- Schema-awareness
- Support for regular expression processing
- Better manipulation of strings
- Advanced grouping possibilities

Our upconversion of the reviews mostly makes use of regular expression processing and string manipulation. Furthermore we use named templates and functions.

For a real good example how to get the best out of XSLT 2.0 when it comes to upconversion please read Michael Kay's article.

Processing the corpus - XSLT 2.0

A short example for the massive cleanup the stylesheet performs

The following snippet is taken from the extensive `main` template

- First we generate the variable `genreTemp` which holds the string with information about the genre of the game

```
<xsl:variable name="genreTemp">
 <xsl:choose>
  <!-- new type -->
  <xsl:when
test="/descendant::table[3]/descendant::td[2]/descendant::div[contains(.,'GEN')]">
   <xsl:analyze-string
select="/descendant::table[3]/descendant::td[2]/descendant::div[contains(.,'GEN')]"
      regex="GENRE:\s(.*)\sSPIEL">
    <xsl:matching-substring>
     <xsl:value-of select="regex-group(1)"/>
    </xsl:matching-substring>
   </xsl:analyze-string>
  </xsl:when>
  <!-- old type -->
   <xsl:otherwise>
    <xsl:value-of
select="/descendant::table[1]/descendant::font[contains(.,'GEN')]/following::i[1]"/>
   </xsl:otherwise>
 </xsl:choose>
</xsl:variable>
```

Processing the corpus - XSLT 2.0

A short example for the massive cleanup the stylesheet performs

The following snippet is taken from the extensive `main` template

- First we generate the variable `genreTemp` which holds the string with information about the genre of the game
- Next we test this string against regular expressions to assign the respective value from the enumerated list

```
<xsl:when test="matches($genreTemp, 'A[\w\.\s]*Adv')">
 <xsl:attribute name="genre">Action-Adventure</xsl:attribute>
</xsl:when>
<!-- [...] -->
<xsl:when test="matches ($genreTemp, '[sS]port|[bB]all|board|Golf|Box|[hH]ock|[tT]enn|
Wrest')">
 <xsl:attribute name="genre">Sport</xsl:attribute>
</xsl:when>
<xsl:when test="matches($genreTemp, '[Aa]ction|Hack|[sS]hoot|Ego|Prüg|FPS')">
 <xsl:attribute name="genre">Action</xsl:attribute>
  <xsl:choose>
   <xsl:when test="matches($genreTemp, 'Ego|FPS')">
    <xsl:attribute name="subgenre">First Person Action</xsl:attribute>
   </xsl:when>
   <!-- [...] -->
```

- Because of word inclusions we take advantage of the case differentiation order

Processing the corpus - XSLT 2.0

To find the title of some documents information stored into external documents has to be taken into account

SYSTEM:	GCN - PAL	SCHWIERIGKEIT:	1-6
ENTWICKLER:	Ubi Soft	SECRETS:	JA
GENRE:	Jump'n Run	SPRACHHÜRDE:	Keine
SPIELER:	1-4 Spieler	PREIS:	ca.60 Euro
HANDBUCH:	Mehrsprachig	TERMIN:	Erhältlich
MEMCARD:	8 Seiten		
60Hz Modus:	JA		

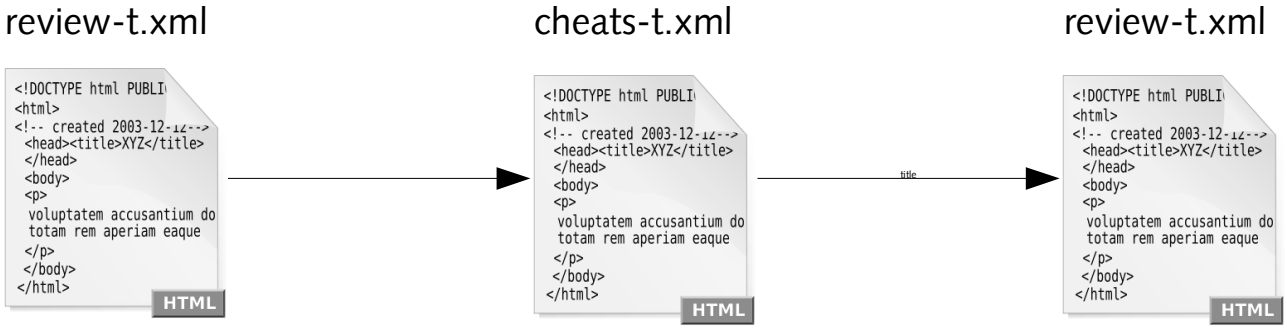
Dieser Testbericht wurde geschrieben von *Matthias Engert* am 24.02.2003

CHEATS: JA | KOMPLETTLÖSUNG: JA | TIME TRIAL / SCORES: JA

Bisher hat uns Ubi Soft ja zumeist mit Spielen anderer Firmen mehr oder weniger beglückt. Eigenentwicklungen waren noch recht wenige darunter. Dabei hat Ubi Soft doch einiges zu bieten. Unter anderem niemand geringeren als Rayman. Einen Charakter den wohl die meisten unter euch kennen und spielerisch schonmal begutachtet haben. Zumindest waren die Rayman Spiele immer recht erfolgreich.

Processing the corpus - XSLT 2.0

To find the title of some documents information stored into external documents has to be taken into account



Here a linked "cheats" or "tips" document is accessed to extract the game title that is hidden in the backlink to the review document

```
<xsl:when test="/descendant::table[2]/descendant::td[1]/div[1]/
 descendant::a[doc-available(concat($filepath,(replace
 (attribute::href, '-i.htm', '-t.xml'))))]">
 <xsl:variable name="doc" select="concat($filepath,replace
  (/descendant::table[2]/descendant::td[1]/div[1]/descendant::a/
  attribute::href, '-i.htm', '-t.xml'))"/>
 <xsl:value-of select="document($doc,.)/descendant::table[1]/descendant::a[1]"/>
</xsl:when>
```

Processing the corpus - XSLT 2.0

- Throughout the transformation many more requirements are met in carrying out the upconversion
- The provided examples are simply illustrative of the process without going into complete detail
- Because the process requires multiple steps and must be applied to many files, we have encapsulated it in XProc

Pipelining with XProc

- XProc a new standard for automating processes like ours through an XML pipeline has been developed by the W3 working group
- XProc has reached status of W3C Recommendation on 11 May 2010
- For our desired all-in-one XML solution, XProc is first choice to handle the pipeline
- There are two implementations for XProc which pass more than 99% of the test suite
  - XML Calabash
  - EMC Documentum XProc Engine (Calumet)
- Our pipeline is tested to work well with XML Calabash Version 0.9.21 (0.9.23 released on 27 July 2010)

Pipelining with XProc

Tasks of the pipeline

- Process the documents that are stored locally in the filesystem recursively
- The pipeline will apply the following tasks to each HTML document

Pipelining with XProc

Tasks of the pipeline

- Process the documents that are stored locally in the filesystem recursively
- The pipeline will apply the following tasks to each HTML document
  - Use HTML Tidy to transform the HTML input into well-formed XML
  - Apply the XSLT script to the output of the former task using an XSLT 2.0 processor
  - Validate the output files according to the XML schema
  - Separate valid from invalid documents

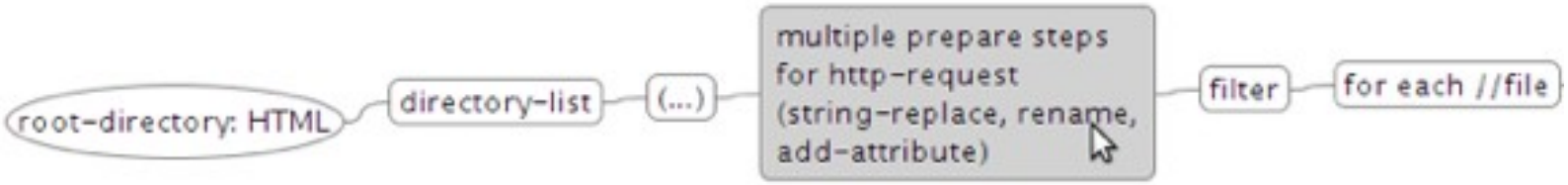
Pipelining with XProc

Tasks of the pipeline

- Process the documents that are stored locally in the filesystem recursively
- The pipeline will apply the following tasks to each HTML document
  - Use HTML Tidy to transform the HTML input into well-formed XML
  - Apply the XSLT script to the output of the former task using an XSLT 2.0 processor
  - Validate the output files according to the XML schema
  - Separate valid from invalid documents
- And finally provide a log of valid documents

Pipelining with XProc

- With XProc XML documents are first class citizens
- We will use HTML as input for our pipeline, this needs a bit precaution
  - We prepared the documents so the encoding of the files is either ISO-8859-1 or UTF-8 and the special characters are masked as numeric entities for the moment. Otherwise there would be encoding errors in the result XML documents
    (we're cheating a bit here, but hey, it's about video games)
  - We take some preparational steps in the pipeline before we can handle HTML (simplified)



Pipelining with XProc - processing multiple HTML files

Let's take a closer look

- First we add the base-uri to get the complete filepath using p:make-absolute-uris.

```
<p:make-absolute-uris match="c:file/@name">
 <p:with-option name="base-uri" select="concat($subdirpath, '/', c:directory/@name)"/>
</p:make-absolute-uris>
<p:string-replace match="c:file/@name" replace="replace(., 'file:', 'file://')" name="replace"/>
<p:rename match="c:file" new-name="c:request"/>
<p:rename match="@name" new-name="href"/>
<p:add-attribute match="c:request" attribute-name="method" attribute-value="get"/>
<p:add-attribute match="c:request" attribute-name="override-content-type" attribute-
value="text/html"/>
```

Pipelining with XProc - processing multiple HTML files

Let's take a closer look

- First we add the base-uri to get the complete filepath using p:make-absolute-uris.
- Then we add slashes using p:string-replace to ensure accordance to the file protocol.

```
<p:make-absolute-uris match="c:file/@name">
 <p:with-option name="base-uri" select="concat($subdirpath, '/', c:directory/@name)"/>
</p:make-absolute-uris>
<p:string-replace match="c:file/@name" replace="replace(., 'file:', 'file://')" name="replace"/>
<p:rename match="c:file" new-name="c:request"/>
<p:rename match="@name" new-name="href"/>
<p:add-attribute match="c:request" attribute-name="method" attribute-value="get"/>
<p:add-attribute match="c:request" attribute-name="override-content-type" attribute-
value="text/html"/>
```

Pipelining with XProc - processing multiple HTML files

Let's take a closer look

- First we add the base-uri to get the complete filepath using p:make-absolute-uris.
- Then we add slashes using p:string-replace to ensure accordance to the file protocol.
- To make sure the file is accessible for the p:http-request step we rename the element and the attribute.

```
<p:make-absolute-uris match="c:file/@name">
 <p:with-option name="base-uri" select="concat($subdirpath, '/', c:directory/@name)"/>
</p:make-absolute-uris>
<p:string-replace match="c:file/@name" replace="replace(., 'file:', 'file://')" name="replace"/>
<p:rename match="c:file" new-name="c:request"/>
<p:rename match="@name" new-name="href"/>
<p:add-attribute match="c:request" attribute-name="method" attribute-value="get"/>
<p:add-attribute match="c:request" attribute-name="override-content-type" attribute-
value="text/html"/>
```

Pipelining with XProc - processing multiple HTML files

Let's take a closer look

- First we add the base-uri to get the complete filepath using p:make-absolute-uris.
- Then we add slashes using p:string-replace to ensure accordance to the file protocol.
- To make sure the file is accessible for the p:http-request step we rename the element and the attribute.
- Furthermore, we need to add the proper attributes for the p:http-request step to work. Since there is no server involved and we do not want to work with binary data, we need to add the attribute override-content-type and attach the value text/html

```
<p:make-absolute-uris match="c:file/@name">
 <p:with-option name="base-uri" select="concat($subdirpath, '/', c:directory/@name)"/>
</p:make-absolute-uris>
<p:string-replace match="c:file/@name" replace="replace(., 'file:', 'file://')" name="replace"/>
<p:rename match="c:file" new-name="c:request"/>
<p:rename match="@name" new-name="href"/>
<p:add-attribute match="c:request" attribute-name="method" attribute-value="get"/>
<p:add-attribute match="c:request" attribute-name="override-content-type" attribute-
value="text/html"/>
```

Pipelining with XProc - processing multiple HTML files

Why are we going the extra mile here?

- If we want to access a single XML document we use p:document
- If we want to access multiple XML documents we can still use p:document
- If we want to access a single HTML document we use p:data
- If we want to access multiple HTML documents we cannot use p:data
  - Why?
  - It is not a step and therefore does not take options
  - It cannot handle variables
- So if we want to access multiple HTML documents we use p:http-request

Pipelining with XProc - processing multiple HTML files

Now we can process the HTML documents in sequence

We use a filter to exclude documents which are not reviews and will not help us to find game titles

These documents may be

- Reader reviews that follow no certain structure
- Hardware reviews
- Or other texts

All other HTML documents will enter the nested loop of the pipeline

Pipelining with XProc

Some technical results

- This pipeline takes approximately half an hour to process the data
- It is relatively independent of CPU speed on an average actual system
- It results in 1573 schema-valid files

Querying

Possible scenarios

- Alternative access
- The wish list

Querying - Alternative access

- The query genres.xq uses two parameters, genre and platform, to search for games of a certain genre on a specific platform by using a collection of all valid XML instance documents
  Here we use genres.xq with the value "Wii" for the platform parameter and the value "Puzzle" supplied for the genre parameter

```
<?xml version="1.0" encoding="UTF-8"?>
<games on="Wii" type="Puzzle">
 <instance score="85" abbreviation="pqwii">Puzzle Quest: Challenge of the Warlords</instance>
 <instance score="80" abbreviation="jewel">Jewel Master: Cradle of Rom</instance>
 <instance score="79" abbreviation="phwwii">Professor Heinz Wolff's Gravity</instance>
 <instance score="76" abbreviation="bbawii">Big Brain Academy </instance>
 <instance score="50" abbreviation="jengawii">Jenga World Tour </instance>
</games>
```

- Since this query was originally developed as an alternative access mechanism, the information returned is very sparse. However, in combination with (X)HTML output containing hyperlinks to the respective review page, it would be sufficient

Querying - Alternative access II

- Sometimes a user searches for games that support certain technical features, such as online content, multiplayer, etc. The techspecs.xq query uses the parameter platform and techspec to retrieve only the reviews of games that include the provided feature.

```
<games on="NDS" featuring="Online">
 <instance score="92" abbreviation="suik">Suikoden Tierkreis </instance>
 <instance score="90" abbreviation="layton">Professor Layton und das geheimnisvolle
Dorf</instance>
 <instance score="89" abbreviation="fesd">Fire Emblem : Shadow Dragon</instance>
 <instance score="88" abbreviation="cpor">Castlevania: Portrait of Ruin</instance>(...)
</games>
```

Querying - Finally: the wish list

how to save the holiday season ;-)

Source: http://www.sheknows.com/sheknows-cares/articles/812363

Querying - Finally: the wish list

Let's say our kid loves racing games, is about 7 years old and gets very annoyed if the game is way too difficult (so we don't want a simulation racing game). It has a Nintendo DS system and we promised a really cool game.

That's where we can make use of our query shoppingList.xq - here we can adjust a few parameters

- The age
- The platform we are interested in
- The genre
- The rating given by the reviewers to exclude garbage
- And finally the max. difficulty

```
XQuery.sh shoppingList.xq age=7 platform=NDS score=70 genre=Rennspiel maxDifficulty=7
```

Querying - Let's see the result of the wish list (abbr.)

```
<games maxAgeRating="7" on="NDS" maxDifficulty="7" type="Rennspiel"
scoreAtLeast="70">
 <instance ageRating="3" score="82" maxDifficulty="7" abbreviation="augt2"
   minDifficulty="1">
  <title>Asphalt Urban GT 2</title>
  <notes>
   <pro>62 Meisterschaften</pro>
   <pro>Für Fans von Arcade Steuerung</pro>
   <pro>Sehr gute Framerate/Technik</pro>
   <pro>Fahrzeugmodelle/Anzahl</pro>
   <pro>Grafische Präsentation</pro>
   <pro>Verschiedene Rennmodi</pro>
   <pro>Motorrad Inhalte</pro>
   <contra>Leichter als der Vorgänger</contra>
   <contra>Polizei in den Meisterschaften</contra>
   <contra>Kein 1C Multiplayer</contra>
  </notes>
 </instance>
 <instance ageRating="3" score="77" maxDifficulty="7" abbreviation="cnr"
minDifficulty="1">
  <title>Cartoon Network Racing</title>
  <notes>
   <pro>Gute Grundsteuerung</pro>
   <pro>Umfangreich duch 4 Cups</pro>
   <pro>Steigende Gegner KI</pro>
   <pro>Lange Strecken</pro>
   <pro>11 gelungene Strecken</pro>
   <pro>Gelungene Items</pro>
   <pro>Viele Belohnungen</pro>
   <pro>Kart Curling Minispiel</pro>
   <contra>Kurventechnik per R-Taste</contra>
   <contra>5 der 16 Strecken</contra>
   <contra>Single Card MP</contra>
   <contra>Zu abruptes Bremsen bei Crashs</contra>
  </notes>
 </instance>
</games>
```

The output is sorted in descending order of the score

Querying - Let's see the result of the wish list (abbr.)

```
<games maxAgeRating="7" on="NDS" maxDifficulty="7" type="Rennspiel"
scoreAtLeast="70">
 <instance ageRating="3" score="82" maxDifficulty="7" abbreviation="augt2"
  minDifficulty="1">
  <title>Asphalt Urban GT 2</title>
  <notes>
   <pro>62 Meisterschaften</pro>
   <pro>Für Fans von Arcade Steuerung</pro>
   <pro>Sehr gute Framerate/Technik</pro>
   <pro>Fahrzeugmodelle/Anzahl</pro>
   <pro>Grafische Präsentation</pro>
   <pro>Verschiedene Rennmodi</pro>
   <pro>Motorrad Inhalte</pro>
   <contra>Leichter als der Vorgänger</contra>
   <contra>Polizei in den Meisterschaften</contra>
   <contra>Kein 1C Multiplayer</contra>
  </notes>
 </instance>
 <instance ageRating="3" score="77" maxDifficulty="7" abbreviation="cnr"
minDifficulty="1">
  <title>Cartoon Network Racing</title>
  <notes>
   <pro>Gute Grundsteuerung</pro>
   <pro>Umfangreich duch 4 Cups</pro>
   <pro>Steigende Gegner KI</pro>
   <pro>Lange Strecken</pro>
   <pro>11 gelungene Strecken</pro>
   <pro>Gelungene Items</pro>
   <pro>Viele Belohnungen</pro>
   <pro>Kart Curling Minispiel</pro>
   <contra>Kurventechnik per R-Taste</contra>
   <contra>5 der 16 Strecken</contra>
   <contra>Single Card MP</contra>
   <contra>Zu abruptes Bremsen bei Crashs</contra>
  </notes>
 </instance>
</games>
```

We provide the pros and cons the reviewers gave

Querying - Let's see the result of the wish list (abbr.)

```
<games maxAgeRating="7" on="NDS" maxDifficulty="7" type="Rennspiel"
scoreAtLeast="70">
 <instance ageRating="3" score="82" maxDifficulty="7" abbreviation="augt2"
  minDifficulty="1">
  <title>Asphalt Urban GT 2</title>
  <notes>
   <pro>62 Meisterschaften</pro>
   <pro>Für Fans von Arcade Steuerung</pro>
   <pro>Sehr gute Framerate/Technik</pro>
   <pro>Fahrzeugmodelle/Anzahl</pro>
   <pro>Grafische Präsentation</pro>
   <pro>Verschiedene Rennmodi</pro>
   <pro>Motorrad Inhalte</pro>
   <contra>Leichter als der Vorgänger</contra>
   <contra>Polizei in den Meisterschaften</contra>
   <contra>Kein 1C Multiplayer</contra>
  </notes>
 </instance>
 <instance ageRating="3" score="77" maxDifficulty="7" abbreviation="cnr"
minDifficulty="1">
  <title>Cartoon Network Racing</title>
  <notes>
   <pro>Gute Grundsteuerung</pro>
   <pro>Umfangreich duch 4 Cups</pro>
   <pro>Steigende Gegner KI</pro>
   <pro>Lange Strecken</pro>
   <pro>11 gelungene Strecken</pro>
   <pro>Gelungene Items</pro>
   <pro>Viele Belohnungen</pro>
   <pro>Kart Curling Minispiel</pro>
   <contra>Kurventechnik per R-Taste</contra>
   <contra>5 der 16 Strecken</contra>
   <contra>Single Card MP</contra>
   <contra>Zu abruptes Bremsen bei Crashs</contra>
  </notes>
 </instance>
</games>
```

But one can think of a con as a pro, or a pro as a con depending on what you want

A perfect holiday season

Source: http://www.ivillage.com/my-kid-cheats-video-games/1-a-69664

Future prospects

- Make the XLST transformation easier to maintain by moving the decision over Type A or Type B document over to XProc
- Rewrite the XSLT transformation
- Use XSD Schema 1.1 assertions to ask for a multiplayer scoring if the game is for more than one player
- Extend the XSD Schema to handle more consoles and their special needs or additional content like tips and tricks or walkthroughs
- Realize web-application using eXist?

Last but not least...

Thank you for your attention!

Contact:

st.haupt@gmail.comlmaik.stuehrenberg@uni-bielefeld.de