

# @ctief Leren

## Het project VISNUE: het kalendersysteem

*Saskia Van de Laer en Adhemar Bultheel*

Werkrapport @L 16, 10 april 2002



Katholieke Universiteit Leuven  
Department of Computer Science  
Celestijnenlaan 200A – B-3001 Heverlee (Belgium)

# @ctief Leren

## Het project VISNUE: het kalendersysteem

*Saskia Van de Laer en Adhemar Bultheel*

Werkrapport @L 16, 10 april 2002

Department of Computer Science, K.U.Leuven

### **Samenvatting**

Dit is de gebruikers- en programmeursgids voor het kalendersysteem ontwikkeld voor het OOI project VISNUE-TWEE (VISualisatie van NUmerieke Experimenten - Toets en Werkomgeving met Evaluatie).

Dit systeem voorziet een kalender waarin de taken en een studietijdsplanning voor het vak kunnen gegeven worden.

Het is een nuttig hulpmiddel voor de student om zijn studie te plannen maar ook voor de docent omdat het een middel is om tot een realistische schatting te komen voor de studiedruk van zijn vak.

# Het project VISNUE: het kalendersysteem

Saskia Van de Laer en Adhemar Bultheel

10 april 2002

<http://www.cs.kuleuven.ac.be/~nalag/research/ALpubs/AL16.abs.shtml>

## Samenvatting

Dit is de gebruikers- en programmeursgids voor het kalendersysteem ontwikkeld voor het OOI project VISNUE-TWEE (VISualisatie van NUmerieke Experimenten - Toets en Werkomgeving met Evaluatie).

Dit systeem voorziet een kalender waarin de taken en een studietijdsplanning voor het vak kunnen gegeven worden.

Het is een nuttig hulpmiddel voor de student om zijn studie te plannen maar ook voor de docent omdat het een middel is om tot een realistische schatting te komen voor de studiedruk van zijn vak.

## 1 Inleiding

Het kalendersysteem is ontwikkeld in het kader van het VISNUE-TWEE project. De bedoeling is dat de studenten per dag of per week opdrachten krijgen zodat dit de hele opzet rond het vak numerieke wiskunde zou ondersteunen, namelijk dat het vak zou geleerd worden tijdens het jaar door permanente oefening, eerder dan als blokvak in de blokperiode te worden verwerkt.

De docent kan gegevens (opdrachten en te besteden tijdsduur) ingeven in het systeem. Dit genereert een via het web raadpleegbare kalender. De student krijgt per week en per dag een overzicht van wat er als leerprestatie verwacht wordt. Het systeem is tegelijk voor de docent een middel om tot een realistische schatting van studiebelasting te komen voor zijn vak. Voor de student is het niet alleen een kalender van taken, maar geeft het gelijktijdig ook een beeld van wat de docent verwacht dat hij doorneemt in de veelheid van aangeboden middelen (cursustekst, lessen, demonstraties, oefeningen, zelftoetsen,...)

## 2 Technische realisatie

Voor de technische realisatie van het kalendersysteem hebben we gekozen voor een XML/XSL beschrijving die toelaat HTML bestanden te genereren.

## 2.1 Wat terminologie

Een overzicht van de verschillende bestanden:

- HTML (HYPERTEXT MARKUP LANGUAGE) is een standaard beschrijving van webpagina's waarbij inhoud en vorm in eenzelfde bestand beschreven worden.
- De XML (EXTENSIBLE MARKUP LANGUAGE) bestanden bevatten een universeel formaat voor gestructureerde webdocumenten en data. Ze beschrijven wat er inhoudelijk in de bestanden moet komen te staan.  
Zie <http://www.w3.org/XML/>.
- De XSL (EXTENSIBLE STYLESHEET LANGUAGE) bestanden beschrijven hoe dat deze XML inhoud moet geformateerd worden. (vergelijkbaar met CSS (CASCADING STYLE SHEETS) voor HTML).  
Zie <http://www.w3.org/Style/XSL/>
- Een DTD (Document Type Definition) bestand beschrijft de valabele bouwstenen van een XML bestand. Daardoor wordt de gehele documentstructuur bepaald.
- Een PL bestand is een bestand met Perl (PRACTICAL EXTRACTION AND REPORT LANGUAGE) script.
- TXT bestanden bevatten gewoon tekst.
- CSH bestanden bevatten C-Shell scripts voor UNIX machines.

De HTML bestanden worden aangemaakt met standaard java programma's `xalan.jar` en `xerces.jar`.

- **xalan** is een XSLT processor die XML bestanden transformeert in HTML, tekst of andere XML bestanden.  
Voor meer informatie zie <http://xml.apache.org/xalan-j/>.
- **xerces** is een java parser die geavanceerde XML supporteert.  
Voor meer informatie zie <http://xml.apache.org/xerces-j/>.

## 2.2 Het aanmaken van de nodige uitvoerbare bestanden en bestanden met gegevens

Het hele systeem is in een PL bestand gestopt: `MakeAll.pl`. Het vraagt als invoerparameters:

- Het jaartal waarin de kalender moet beginnen. Dit moet bestaan uit 4 cijfers.
- De maand waarin de kalender moet beginnen. (1-12)
- Het aantal maanden dat de kalender minimaal moet bevatten. (1-12)  
De huidige implementatie realiseert de kalender als een tabel waarin AMPTR maanden per rij worden gerealiseerd. Voor de layout van de pagina wordt de kalender aangevuld tot een veelvoud van AMPTR maanden.
- Optioneel wordt een vierde parameter AMPTR gegeven die aangeeft hoeveel maanden per rij men in de kalender wil tonen. De verzuimwaarde is 3.

Bijvoorbeeld `MakeAll.pl 2001 9 7` zal de bestanden klaarmaken om de maanden september tot december 2001 en januari tot mei 2002 te genereren.

Het uitvoeren van `MakeAll.pl` resulteert in een aantal bestanden:

- `inv.pl`: een perlscriptje dat verder zal gebruikt worden.
- `kal.dtd`: de DTD beschrijving van een jaar: Een jaar bestaat uit weken, een week bestaat uit dagen, een dag bestaat uit taken, een taak bestaat uit een beginuur, een einduur en een duur. Indien die uren ontbreken wordt een verzuimwaarde ingevuld. Sommige vlottende taken, worden ook aan weken toegekend i.p.v. aan dagen. Het bestand `kal.dtd` is in appendix gegeven.
- `kal.xml`: de XML beschrijving van de kalender.
- `dag.xsl`, `week.xsl`, `jaar.xsl`: de XSL beschrijving van de respectievelijke onderdelen.
- Voor elk week een bestand `week<weeknummer>.xml` in de map `week<weeknummer>`; als het weeknummer meerdere keren voorkomt in de kalender (dit kan als de kalender 12 kalendermaanden bevat en willekeurig in het jaar begint), heet de 2e map en het 2e xmlbestand `week<weeknummer>b` resp. `week<weeknummer>b.xml`.
- Voor elke week een bestand `Makefile` in de map `week<weeknummer>`; ook hier geldt dat indien er 2 weken zijn met hetzelfde weeknummer, de 2e `Makefile` in de map `week<weeknummer>b` staat.
- `start.csh`, `test.csh`: C-shell scriptjes, waarvan het gebruik later wordt beschreven.

Zo een typisch xmlbestand voor bijvoorbeeld de eerste week van het jaar, `week01/week01.xml`, waarin niets speciaal moet gedaan worden ziet er als volgt uit:

```
<?xml version="1.0" encoding="UTF-8"?>
<week number="01" startmonth="01" endmonth="01">
<day number="31" month="12">
</day>
<day number="01" month="01">
</day>
<day number="02" month="01">
</day>
<day number="03" month="01">
</day>
<day number="04" month="01">
</day>
<day number="05" month="01">
</day>
<day number="06" month="01">
</day>
</week>
```

Het XML-document met de gegevens moet voldoen aan een welbepaalde structuur, waarin een jaar bestaat uit weken, weken uit dagen en per dag kunnen er geen, 1 of meer items op de planning staan zoals beschreven in het DTD bestand `kal.dtd`.

Een typische `Makefile` ziet eruit als volgt

```
XMLFILES=week01.xml
XMLKAL=./kal.xml
DTD=./kal.dtd
XSLDAG=./dag.xsl
XSLWEEK=./week.xsl
XSLJAAR=./jaar.xsl
MAKEF=Makefile
all : week.htm ../kal.htm \
    ov1231.htm ov0101.htm ov0102.htm ov0103.htm ov0104.htm ov0105.htm ov0106.htm
```

```

week.htm : $(XMLKAL) $(XSLWEEK) $(XMLFILES) $(MAKEF)
java -classpath ../xerces.jar:../xalan.jar org.apache.xalan.xslt.Process \
    -in ../kal.xml -xsl ../week.xsl -PARAM weeknr 01 -out week.htm
../kal.htm : $(XMLKAL) $(XSLJAAR) $(XMLFILES) $(MAKEF)
java -classpath ../xerces.jar:../xalan.jar org.apache.xalan.xslt.Process \
    -in ../kal.xml -xsl ../jaar.xsl -out ../kal.htm
ov1231.htm : $(XMLKAL) $(XSLDAG) $(XMLFILES) $(MAKEF)
java -classpath ../xerces.jar:../xalan.jar org.apache.xalan.xslt.Process \
    -in ../kal.xml -xsl ../dag.xsl -PARAM maandnr 12 -PARAM dagnr 31 -out ov1231.htm
ov0101.htm : $(XMLKAL) $(XSLDAG) $(XMLFILES) $(MAKEF)
java -classpath ../xerces.jar:../xalan.jar org.apache.xalan.xslt.Process \
    -in ../kal.xml -xsl ../dag.xsl -PARAM maandnr 01 -PARAM dagnr 01 -out ov0101.htm
ov0102.htm : $(XMLKAL) $(XSLDAG) $(XMLFILES) $(MAKEF)
java -classpath ../xerces.jar:../xalan.jar org.apache.xalan.xslt.Process \
    -in ../kal.xml -xsl ../dag.xsl -PARAM maandnr 01 -PARAM dagnr 02 -out ov0102.htm
ov0103.htm : $(XMLKAL) $(XSLDAG) $(XMLFILES) $(MAKEF)
java -classpath ../xerces.jar:../xalan.jar org.apache.xalan.xslt.Process \
    -in ../kal.xml -xsl ../dag.xsl -PARAM maandnr 01 -PARAM dagnr 03 -out ov0103.htm
ov0104.htm : $(XMLKAL) $(XSLDAG) $(XMLFILES) $(MAKEF)
java -classpath ../xerces.jar:../xalan.jar org.apache.xalan.xslt.Process \
    -in ../kal.xml -xsl ../dag.xsl -PARAM maandnr 01 -PARAM dagnr 04 -out ov0104.htm
ov0105.htm : $(XMLKAL) $(XSLDAG) $(XMLFILES) $(MAKEF)
java -classpath ../xerces.jar:../xalan.jar org.apache.xalan.xslt.Process \
    -in ../kal.xml -xsl ../dag.xsl -PARAM maandnr 01 -PARAM dagnr 05 -out ov0105.htm
ov0106.htm : $(XMLKAL) $(XSLDAG) $(XMLFILES) $(MAKEF)
java -classpath ../xerces.jar:../xalan.jar org.apache.xalan.xslt.Process \
    -in ../kal.xml -xsl ../dag.xsl -PARAM maandnr 01 -PARAM dagnr 06 -out ov0106.htm

```

Het betreft een klassieke beschrijving van afhankelijkheden en instructies voor het aanmaken van de HTML bestanden voor deze week, de dagen in deze week en de ganse kalender.

Alle eventuele oude bestanden worden verwijderd tijdens de uitvoering van `MakeAll.pl`, behalve het takenbestand (zie de betreffende paragraaf 2.4).

## 2.3 Het aanmaken van de html-bestanden

Alles staat nu klaar om de kalender te genereren. Dat gebeurt door de volgende stappen opeenvolgend uit te voeren.

- Klaarmaken van een tekstbestand met beschrijving van de taken en de dag of periode dat ze moeten worden uitgevoerd. Dit beschrijven we in de volgende paragraaf.
- Uitvoeren van het script `inv.pl`: Dit script vraagt als invoer een tekst bestand waarin beschreven staat welke taken er wanneer moeten uitgevoerd worden door de studenten. Hoe dit tekstbestand er moet uitzien beschrijven we in de volgende paragraaf. Het resultaat is dat deze taken worden ingevuld in de gepaste bestand `week<weeknr>/week<weeknr>.xml`, ofwel als item bij een bepaalde dag, ofwel als item voor de week (achteraan in het element week).

## 2.4 De beschrijving invullen, aanpassen en verwijderen van taken.

Het tijdstip waarop een taak moet uitgevoerd worden kan 3 formaten hebben.

- *op een bepaalde dag.*  
Hiervoor geef je op  
dd/mm (met dd = dag in 2 cijfers, en mm = maand in 2 cijfers)

- op een vaste dag van de week gedurende een aantal weken.

Hiervoor geef je op

`wd weekreeks` (met `wd = ma|di|wo|do|vr|za|zo`) en `weekreeks` een lijst van weken door komma's gescheiden. Begin en einde van elke groep opeenvolgende weken wordt aangeduid door het weeknummer (week in het burgerlijk jaar). Begin- en eindweek wordt gescheiden door een koppelteken; het weeknummer voor het koppelteken moet kleiner dan of gelijk aan het weeknummer na het koppelteken zijn. Eventueel kan het eindweeknummer weggelaten worden indien dit hetzelfde is als het beginweeknummer. Bijvoorbeeld `ma 7-11` of `do 7-11,13,16-18,20`.

- in een zekere periode, maar niet op een bepaalde dag.

Hiervoor geef je op

`dd/mm-dd/mm` (met `dd = een dag`, en `mm = een maand`. Steeds 2 cijfers, dus 02 en niet 2.)

Het formaat voor een taak-moment is als volgt: `@ITEM beginuur einduur duur` of `@ITEM duur einduur`. Hierin is

- `@=[*|!]`, dus ofwel `*` ofwel `!` ofwel ontbreekt het.

- Gebruik een `*` om deze taak toe te voegen
- Gebruik een `!` om deze taak te verwijderen
- Gebruik geen teken om deze taak ongewijzigd te laten

- Vervolgens geef je `beginuur`, `einduur` en `duur` op.

- `beginuur` en `einduur` zijn van de vorm `hh.mm`. `hh` is een getal tussen 0 en 23; `mm` zijn minuten en worden steeds met 2 cijfers aangegeven.
- `duur` is de studiebelasting in gehele getallen (uren) uitgedrukt.
- Indien een tijdstip niet relevant is dan kan dat vervangen worden door een koppelteken. Enkel de vermelde gegevens worden getoond. De vorm `@ITEM duur einduur` wordt gebruikt voor een vlottende taak waar enkel het einduur van belang is, bijvoorbeeld een deadline voor een practicum.

Daarna volg één regel met de eigenlijke omschrijving van de taak. De hele taak moet op 1 regel beschreven worden zonder dat deze speciale tekens (andere dan alfanumerische tekens) (zie Appendix B) of geleinden bevat. Uitzonderingen hierbij zijn: `<`, `>` en `"`. HTML elementen kunnen niet ingevoerd worden tenzij een HTML-link.

Taken op dezelfde datum (of zelfde weekdag en weekreeks, of zelfde periode) kunnen gegroepeerd worden onder 1 hoofding

Voorbeeld van een takenbestandje:

```
01/01
*ITEM 11.00 - 1
testitem1 <A HREF="http://www.kuleuven.ac.be">bepaalde dag</A>
ITEM 10.00 12.00 2
testitem2 <A HREF="http://www.kuleuven.ac.be">bepaalde dag</A>Test
01/01-04/01
*ITEM 4 12.00
testitem3 periode
WO 1-3,5
!ITEM 10.00 11.00 1
testitem4 weekdag in weken
```

Er kunnen blanco lijnen voorkomen in het gegevensbestand, die worden gewoon genegeerd.

## 2.5 Genereren van de eigenlijke html code

Men kan dit op verschillende manieren doen.

- Ofwel weet men welke week men moet hergenereren, en dan kan men in de betreffende week-folder `make` uitvoeren, of eventueel zelfs `make week.htm`.
- Ofwel start men het hele proces om alle html bestanden voor alle dagen, voor alle weken en voor de hele kalender te genereren. Dat gebeurt door het uitvoeren van het shell script `start.csh`. Zolang dit script blijft draaien zal er getest worden of bepaalde bestanden gewijzigd werden om eventueel nieuwe html code te genereren.

## 2.6 Opnieuw aanmaken van de kalender voor een andere periode

In dit geval zal men moeten volgende stappen uitvoeren:

1. Stop indien nodig het scriptje `start.csh`.
2. Uitvoeren van `MakeAll.pl` en alle volgende stappen als hierboven beschreven. Alle bestanden worden opnieuw aangemaakt en overschrijven de eventueel nog bestaande bestanden.

## A Appendix A: Het kal.dtd bestand

```
<!-- Het rootelement dat tags van alle types die apart -->
<!-- kunnen voorkomen, kan bevatten. -->
<!-- Academiejaar. -->
<!ELEMENT year (week)*>
<!-- Eerste jaartal van het academiejaar. -->
<!ATTLIST year number      CDATA #REQUIRED>
<!-- Tweede jaartal van het academiejaar. -->
<!ATTLIST year endnb      CDATA #REQUIRED>

<!-- Week in het jaar. -->
<!ELEMENT week (day | item)*>
<!-- Weeknummer. -->
<!ATTLIST week number      CDATA #REQUIRED>
<!-- Maand waarin de week begint (meestal behoort de volledige week tot 1 maand). -->
<!ATTLIST week startmonth  CDATA #REQUIRED>
<!-- Maand waarin de week eindigt (meestal dezelfde als de beginmaand). -->
<!ATTLIST week endmonth   CDATA #REQUIRED>

<!-- Kalenderdag. -->
<!ELEMENT day (item)*>
<!-- Kalenderdag: nummer. -->
<!ATTLIST day number      CDATA #REQUIRED>
<!-- Maand waartoe deze dag behoort. -->
<!ATTLIST day month      CDATA #REQUIRED>

<!-- Taak. -->
<!ELEMENT item EMPTY>
<!-- Beschrijving van de taak. -->
<!ATTLIST item description CDATA #REQUIRED>
<!-- Beginuur; verzuimwaarde 99.999 -->
<!ATTLIST item starthour  CDATA "99.999">
<!-- Einduur; verzuimwaarde 99.999 -->
```



```
<!ATTLIST item endhour      CDATA "99.999">
<!-- Duur / geschatte belasting in uren; verzuimwaarde is 9999 -->
<!ATTLIST item duration     CDATA "9999">
<!-- Periode waarin de taak uitgevoerd dient te worden (mmdd-mmdd). -->
<!-- Optioneel: enkel voor taken die niet gebonden zijn aan 1 specifieke dag. -->
<!-- Deze taken horen te staan bij een week i.p.v. bij een dag. -->
<!ATTLIST item period       CDATA #IMPLIED>
```

## B Appendix B: speciale tekens.

Met speciale tekens worden hier bedoeld: tekens die op het toetsenbord staan maar die als HTML-entiteit moeten ingevoerd worden.

Voor een lijst met alle HTML-entiteiten, zie <http://www.w3.org/TR/html401/sgml/entities.html>.

Volgende tekens worden automatisch correct omgezet:

- het kleiner-dan-teken (<).
- het groter-dan-teken (>).
- dubbele aanhalingstekens (“).

Volgende tekens op het toetsenbord moeten ingegeven worden als HTML-entiteiten:

- de ampersand (&): &amp;
- het deelteken: &uml;
- ^ : &circ;
- de tilde: &tilde;
- ü: &uuml;
- é: &eacute;
- ê: &ecirc;
- è: &egrave;
- mu ( $\mu$ ): &mu;
- het Euroteken: &euro;