

# Bijlage 1

## CPAN6 projectvoorstel

Mark A.C.J. Overmeer\*  
Arnhem

### Doel van het project

CPAN6 is een nieuw soort dienst op Internet; voor zover bekend is er geen vergelijkbare applicatie beschikbaar. CPAN6 bestaat alleen nog als ontwerp, in de laatste versie verkrijgbaar via <http://cpan6.org>. Doel van dit project is om een implementatie van het ontwerp te realiseren, en tegelijkertijd de groei van een gebruikersgemeenschap te stimuleren.

### Korte beschrijving

Bij het gebruik van computer-systemen en internet zijn data-verzamelingen heel normaal. Een paar voorbeelden:

- een directory (*map*) met files
- een database
- een verzameling digitale foto's
- een verzameling Open Source software modules, zoals bijeengebracht in Perl's CPAN archief<sup>1</sup>
- project referenties in sourceforge<sup>2</sup> en Freshmeat<sup>3</sup>.

---

\*e-mail: [mark@overmeer.net](mailto:mark@overmeer.net), web-site: <http://solutions.overmeer.net>

<sup>1</sup>CPAN: <http://cpan.org>

<sup>2</sup>Sourceforge: <http://sourceforge.net>

<sup>3</sup>Freshmeat: <http://freshmeat.net>

- wikipedia, een verzameling artikelen

Op sommige plekken zien we verzamelingen van deze verzamelingen:

- een home-directory groepeert iemands foto-verzameling, de email, de privé ontwikkelde software en een aantal office documenten gemaakt voor het werk.
- een ftp-server mirrort de inhoud van verschillende andere ftp-servers
- CVS/GIT/SVN archieven met daarin de versie-historie van meerdere projecten.

In elk van deze 'meta-verzamelingen' worden de grenzen tussen de oorspronkelijke bronnen opgeheven. En dat is nu precies waarin CPAN6 verschilt van de bestaande infrastructures: CPAN6 bakent de grenzen van een verzameling data (een 'archief') af, en maakt er een geheel van.

CPAN6 vervult de volgende taken:

- het legt de spelregels van het archief vast: wie is verantwoordelijk, wie mag bijdragen, wie mag ophalen, welke namen mogen worden gebruikt, etc
- verzorgt de distributie van de inhoud van het archief (mirroring)
- leidt de gebruiker naar de optimale mirror voor een archief
- beheert de relaties tussen archieven en materiaal in de archieven
- maakt het beheren van meerdere archieven op één systeem makkelijk
- helpt bij het terugvinden van data in een archief
- bewaakt de integriteit van de data
- beveiligt van het data-transport (trust) vanaf de persoon die de data aanbiedt totaan diegene die de data gaat gebruiken.
- legt de licentie-bepalingen vast.

Bij bestaande data-verzamelingen worden de bovenstaande activiteiten ad-hoc geregeld; CPAN6 maakt dit expliciet. CPAN6 is onafhankelijk van het type verzamelde informatie, het concentreert zich op het beheer ervan.

## Voorbeelden van toepassingen

Het ligt in de bedoeling van dit project om –naast de implementatie van de basis software– enkele applicaties te demonstreren (en liefst ook gebruikt te krijgen in bestaande applicaties). Dit zou bijvoorbeeld kunnen zijn:

- een CPAN6 toegang tot de software modules voor de programmeertaal Perl, zoals die aan worden geboden via het archief CPAN.
- CPAN6 als beheers- en beveiligingsschil over ftp-servers.
- het delen van het gezamenlijk foto-materiaal van personen binnen een huishouden met meerdere computers.

## 1 Status CPAN6 Ontwikkeling

CPAN6 wordt momenteel ontwikkeld door Mark Overmeer<sup>4</sup> met Sam Vilain<sup>5</sup> als belangrijkste sparing partner. Mark heeft momenteel rond de drie maanden eigen tijd in het project gestoken. Sam besteedde een week of twee van zijn tijd, voornamelijk aan brainstorm sessies.

### 1.1 Activiteiten

Tot nu toe zijn de volgende activiteiten ondernomen:

- drie ontwerp-documenten: een overzicht/inleiding, een ontwerp voor de infrastructuur (CPAN6) en een ontwerp voor een archief-beheerder (Pause6). Gezamenlijk ruim 100 pagina's.
- een website met de project-documentatie <http://cpan6.org>, en een mailinglist
- een namespace voor de ontwikkelde XML-schema's op <http://schemas.cpan6.net>.
- 90 minuten tutorial tijdens de YAPC::EU 2006<sup>6</sup> met Perl ontwerper Larry Wall als toehoorder.

---

<sup>4</sup>Mark Overmeer is ZZPer, <http://solutions.overmeer.net>

<sup>5</sup>Sam Vilain is Open Source ontwikkelaar van Catalyst Risk Management in Nieuw Zeeland, <http://catalyst.co.nz>

<sup>6</sup>YAPC::EU is de jaarlijkse bijeenkomst van Europese Perl programmeurs, 20 aug – 1 sept 2006 in Birmingham UK.

- 3 minuten presentatie tijdens de HOSP<sup>7</sup> nieuwjaarsborrel 2007.
- tijdens de GPW 2007<sup>8</sup> werd een 180 minuten durend tutorial worden gegeven.
- tijdens de Perl Hackathon 2007<sup>9</sup> werden mensen betrokken bij het maken technologie-keuzes.
- tijdens de DPW 2007<sup>10</sup> is 45 minuten ingeruimd voor CPAN6.

## 1.2 Randvoorwaarden

Bij de implementatie worden de volgende spelregels worden gehanteerd:

- overal wordt gebruikgemaakt van Unicode
- de user-interfaces worden onmiddellijk gebouwd op meertaligheid
- de code wordt platform-onafhankelijk geschreven
- waar mogelijk wordt gebruik gemaakt van bestaande (Open Source) applicaties
- indien beschikbaar worden (ingewikkelde) in RFCs of door ISO/W3C beschreven protocollen gebruikt in plaats van (“gemakkelijke”) eigen specificaties
- van elk component wordt minimaal één implementatie gemaakt, met de mogelijkheid alternatieven toe te voegen<sup>11</sup>
- componenten in het netwerk moeten stuk voor stuk software en technologie-upgrades kunnen doorvoeren
- de behoefte tot upgrade van client-side applicaties wordt geminimaliseerd, om gebruikers optimaal gemak te geven.

Bovenstaande randvoorwaarden zijn ambitieus maar haalbaar.

---

<sup>7</sup>HOSP: Holland Open Source Platform, <http://hosp.nl>

<sup>8</sup>GPW: German Perl Workshop, <http://perlworkshop.de>, 21-23 februari in München.

<sup>9</sup>2-4 maart in Arnhem

<sup>10</sup>DPW: Dutch Perl Workshop, <http://perlworkshop.nl>, 16 maart in Arnhem.

<sup>11</sup>Bijvoorbeeld, data uitwisseling via HTTP wordt gerealiseerd, maar dusdanig opgezet dat rsync, SOAP en FTP alternatieven eenvoudig kunnen worden toegevoegd.

### 1.3 Technologie keuzes

In de eerste maanden van 2007 is studie gedaan naar de technieken die gebruikt gaan worden bij de realisatie van het ontwerp. De volgende keuzes zijn daaruit voortgekomen, voor de eerste implementatie:

- Alle data-structuur definities zijn vastgelegd in XML Schema's; niet omdat deze syntax ideaal is, maar omdat schema's zijn uitgegroeid tot een industrie-standaard.
- De code wordt geschreven in Perl, vanwege de grote platform-onafhankelijkheid en -ondersteuning, het grote aantal bestaande software-componenten en de haalbare ontwikkelsnelheid.
- Als grafisch user-interface zal Web 2.0<sup>12</sup> worden gebruikt. Deze techniek staat zeer in de belangstelling; steeds meer complexe applicaties worden hierop gebaseerd, zoals "Google Docs & Spreadsheets". Veel ontwikkelaars zijn hiermee bekend, waardoor de kans op bijdragen van derden voor dit onderdeel van CPAN6 het grootst is.
- De data definities zijn beschreven in XML Schema's. Logisch is daarom om, waar nodig, voor databases te kiezen die XQuery ondersteunen. Voor CPAN6 servers zal daarom in eerste instantie MonetDB worden gebruikt. Archieven kunnen ook zonder database worden gebruikt (om toepassing simpeler te maken), maar waarschijnlijk met minder zoek-faciliteiten of sterk verminderde performance.

### 1.4 Voorbereiding implementatie

Er is nog geen begin gemaakt met de implementatie van de specifieke software, maar er zijn wel enkele basis-modules in ontwikkeling die zeker gebruikt gaan worden. Omdat de initiële CPAN6 implementatie zal geschieden in Perl, zijn dit dus Perl modules. Deze modules kunnen ook buiten dit project gebruikt worden.

**OODoc** is een uitbreiding op het Perl's standaard documentatie systeem POD. OODoc maakt het mogelijk om de gebruikersinformatie van

---

<sup>12</sup>De term "Web 2.0" staat voor een scala aan website technieken die meer interactie tussen de client en server mogelijk maken; vooral JavaScript en AJAX. De kwaliteit van een web-interface komt daardoor dicht bij de mogelijkheden van andere grafische interfaces, met als voordelen vooral locatie-onafhankelijkheid en platform onafhankelijkheid. Gekozen is voor de MIT/GPL gelicentieerde bibliotheek jQuery voor basis interface componenten.

grote projecten met veel componenten tot één heterogene beschrijving samen te smelten. Deze module bestaat al een aantal jaren, maar zal worden uitgebreid.

**XML::Compile** kan Perl data-structuren kan vertalen tot perfect XML en weer terug, op basis van de gestandaardiseerde XML schema's. Er bestonden al een dertigtal XML modules voor Perl, maar die waren allemaal kwalitatief ongeschikt. Deze module is recent, en wordt momenteel uitgebreid met WSDL en SOAP functionaliteit.

**Sys::Indep** versimpelt het werk om platform onafhankelijk (*System Independent*) software te ontwikkelen met Perl. Een grote hoeveelheid bestaande Perl modules worden vervangen door één overzichtelijke implementatie. Deze module is nog niet vrijgegeven.

**XML::Compile::UI** (?) CPAN6 kan alleen een succes worden als het voor gebruikers prettig te benaderen is. Helaas zal dat betekenen dat er verschillende soorten interfaces moeten worden gemaakt voor overlappende taken. In ieder geval moet er een command-line interface zijn voor scripting, een curses interface en een grafisch interface voor UNIXen en Windows.

Gedacht wordt aan een automatische vertaler van de gebruikte XML schema's naar UI (User Interface) basis-elementen. De schema-parser van `XML::Compile` kan hiertoe waarschijnlijk worden uitgebreid. Deze elementen moeten daarna per type interface op een andere manier ge-layout en ingezet worden. Het GNU gettext systeem voor taal-onafhankelijkheid zit hierin geïntegreerd.

## 2 Wensen

De volgende stappen staan op het plan voor versie 1.0:

1. met enige spoed moet er een proto-type worden gemaakt om naast het ontwerp (wat abstract en groot is) een *hands-on* demonstratie (wat simpel zou moeten zijn) te kunnen laten zien.
2. het is nodig om structureel tijd te kunnen besteden aan het creëren van interesse in het product. Daarvoor moet actief contact worden gezocht met ontwikkelaars van allerlei soorten verzamelingen. Daarnaast is het noodzakelijk om artikelen te schrijven en regelmatig presentaties te geven.

3. de basis-bouwstenen moeten verder worden uitgewerkt en aangepast het doel (de CPAN modules uit het voorgaande hoofdstuk).
4. een implementatie van de volgende gebruiks-modes van CPAN6:
  - (a) ontwikkelaar in eigen directory (daemon-loos)
  - (b) gebruik vanaf CD/disk (off-line)
  - (c) beheer op systeem-niveau (stand-alone)
  - (d) replicatie in een netwerk (met mirrors)
5. het ontwikkelen van verschillende soorten user-interfaces om bovenstaande functies voor de gebruiker makkelijk toegankelijk te maken.
6. demonstratie producten
  - (a) beheren van eigen software releases en/of foto's
  - (b) beheer- en beveiligingsschil over ftp-servers
  - (c) parallel toegang tot de software modules uit Perl's bestaande CPAN archief

### **3 Ontwikkelingsvoorstel**

Het ontwerp van CPAN6 is al ver gevorderd. Daarom zal het komende jaar de focus komen te liggen op de implementatie en publieke acceptatie van de CPAN6 plannen.

#### **3.1 Implementatie**

De implementatie van CPAN6 worden geschat op 6 man-maanden werk: het beslaat wel veel componenten aan de zichtbare buitenkant, maar de interne structuur van CPAN6 is heel regelmatig wat implementatie zeer versnelt.

De werkzaamheden zullen uitgespreid worden over minimaal een jaar, waardoor onderdelen tijd krijgen om te stabiliseren. De componenten worden zoveel mogelijk multi-functioneel gemaakt. Toepassingen van dezelfde componenten door andere personen in andere applicaties zorgt voor een toename in 'volwassenheid' van het onderdeel.

User-interfaces kosten verreweg de grootste inspanning voor de realisatie.

De CPAN6 faciliteiten worden aangeboden via een grafisch, een scriptbaar en een SOAP interface.

Verwacht wordt dat het pas mogelijk is om toegewijde mede-ontwikkelaars te krijgen vanaf het moment dat enkele aantrekkelijke user-interface componenten demonstreerbaar zijn. Dit geldt ook voor mensen die al in het voortraject betrokken zijn geweest (zoals voorgenoemde Sam Vilain en de Perl gebruikersgemeenschap): de uitgebreidheid van CPAN6 schrikt potentiële ontwikkelaars momenteel nog te veel af.

### **3.2 Promotie en Toepassing**

Het lastigste onderdeel van elk nieuw product is de promotie: in dit geval het stimuleren van toepassingen. Als voordeel heeft CPAN6 dat het op velerlei gebied kan worden ingezet. Het is dus ook helemaal niet zeker dat de mensen uit de Perl hoek (waar de basis voor het plan ligt) als eerste over zullen gaan tot acceptatie van de CPAN6 infrastructuur.

Om een succes te worden zal veel aan promotie moeten worden gedaan: gepland is gemiddeld vier uur per week. Hier valt onder:

- per e-mail contact zoeken met potentiële gebruikers; ontwikkelaars van Open Source software die CPAN6 als verzamelaar kunnen gebruiken.
- het maken van presentatie- en tutorial-materiaal
- het geven presentaties en tutorials op conferenties, vooral in Europa.
- het organiseren van BoFs op deze conferenties
- het bijhouden van de website op niet-technisch gebied: logo, layout, voorbeelden en FAQs
- deelname aan (of organisatie van) hackathons om tot toepassing te komen
- publieke evaluatie van de user-interfaces

De inspanning op het gebied van promotie zal gemiddeld minimaal 4 uur per week bedragen. Door zelf alleen presentaties in Europa te geven worden kosten en vooral op werkbare uren bespaard: anders blijft er te weinig tijd over voor implementatie.



### 3.3 Tijdsplanning

In deze fase is nog geen gedetailleerde tijdsplanning te maken. De snelheid waarmee het project zich kan ontwikkelen heeft onder andere te maken met het succes met binnenhalen van sponsorgelden voor het gedeelte wat niet door NLnet wordt gesteund.

Als globale tijdplanning wordt momenteel uitgegaan van een prototype ter demonstratie beschikbaar in juli, en versie 1.0 (met de functionaliteit beschreven in vorig hoofdstuk) in maart 2008.

### 3.4 Begroting

Uitgaande van een uur-tarief van €50, komt de financiële behoefte voor de implementatie uit op 6 maanden  $\times$  160 uur per maand  $\times$  €50 per uur, totaal €48k.

Voor de promotie en ontwikkeling toepassingsgebieden tot aan versie 1.0 in maart 2008 staat gepland 4 uur per week  $\times$  50 weken  $\times$  €50 per uur, totaal €10k.

Reis- en verblijfskosten zijn moeilijk te overzien. Bijkomende kosten zoals apparatuur, domeinnamen, etc zijn relatief klein. Gezamenlijk geschat op €3k.

Hiermee komt het totaal van geplande kosten tot aan de release van versie 1.0 per maart 2008 op €60k.

### 3.5 Bijdrage NLnet

NLnet heeft aangegeven maximaal €30k bij te dragen aan dit project. Door deze bijdrage wordt het realistisch om met de implementatie te beginnen.

Gezien de doelstelling van NLnet wordt deze bijdrage als volgt een bestemming gegeven:

- alle geplande tijd ter promotie en ontwikkeling van gebruikersgemeenschappen (inclusief optredende de reiskosten) tot aan versie 1.0, verwacht rond 1 maart 2008.
- de eerste twee man-maanden van de ontwikkeling, gespreid uitgevoerd over vier kalender maanden, dus tot 1 juli 2007. Op dat mo-

ment zou een werkend, realistisch prototype demonstreerd moeten kunnen worden.

Volgens de berekening van de begroting in vorige paragraaf komt dit geheel uit op ongeveer €33k exclusief de reiskosten.

Om verantwoording af te leggen over de voortgang van het project zal verslag worden gedaan:

- rond 1 mei; na ruim twee kalendermaanden,
- rond 1 juli; na vier kalendermaanden, als het eerste ontwikkelingsbudget is verbruikt,
- rond 1 november 2007 en 1 maart 2008; over de promotionele- en toepassingsontwikkelingen.

Declaraties worden verstuurd na goedkeuring van het voortgangsverslag, op elk van de vier rapportage-momenten.