PDF

Tworzenie dynamicznych dokumentów PDF w TEX-u

Piotr Bolek

PDF jest formatem dokumentów elektronicznych, w których można definiować obiekty hipertekstowe, takie jak odnośniki (odsyłacze hipertekstowe), notki i zakładki. Dodatkowe atrakcje takie jak dźwięki, animacje itp. do poprawnego działania wymagają odpowiednich wtyczek (*plug-in*) do programu Acrobat Reader. Jednym z rozszerzeń wymagających specjalnej wtyczki Acrobat Forms jest możliwość definiowania formularzy i programowania z wykorzystywaniem języka JavaScript. Wtyczka Acrobat Forms jest dostępna w standardowym pakiecie programu Acrobat Reader dla wszystkich systemów począwszy od wersji 4.0. Wcześniej dla niektórych systemów można ją było ściągnąć z serwera firmy Adobe. Użytkownik programu Acrobat Reader wyposażonego w taką wtyczkę może używać dokumentów z formularzami w podobny sposób jak w przypadku HTML-a. Dostępne są wszystkie typowe pola: listy, przyciski, pola wyboru (*check/radio boxes*) i pola tekstowe. Dane z formularza mogą być przesłane do serwera WWW i tam przetworzone. Istnieją dwa sposoby przesłania danych z formularza PDF-owego do serwera WWW – w konwencji HTML-owej (CGI) albo w opracowanym przez firmę Adobe formacie FDF.

Mechanizmy udostępniane przez JavaScript-owe rozszerzenie programu Acrobat Reader nie ograniczają się do możliwości tworzenia i obsługi prostych formularzy. Możliwe jest tworzenie dokumentów, które mogą być czymś w rodzaju małych aplikacji. Wartości wpisane w pewne pola mogą być wykorzystane do wyliczenia wartości innych pól, poza tym pola formularzy mogą być używane w kontekście nieco innym, ale o tym za chwilę.

Skrypty JS mogą być dołączane do dokumentów PDF w kilku miejscach. Po pierwsze istnieje możliwość umieszczenia plików z rozszerzeniem `.js` w katalogu, w którym znajduje się plik z wtyczką. Rozwiązanie to jest jednak kiepskie, bo wymaga praw dostępu do katalogu systemowego, co w ogólnym przypadku może być niemożliwe. W samym dokumencie PDF możemy dołączać skrypty w dwóch miejscach: globalnie – wywoływane są one w czasie otwierania dokumentu, albo lokalnie jako kod obsługujący zdarzenie związane z konkretnym polem. Ponieważ skrypty są przechowywane razem z dokumentem, to warto definiować globalne funkcje i odwoływać się do nich w procedurach obsługi zdarzeń związanych z konkretnymi polami.

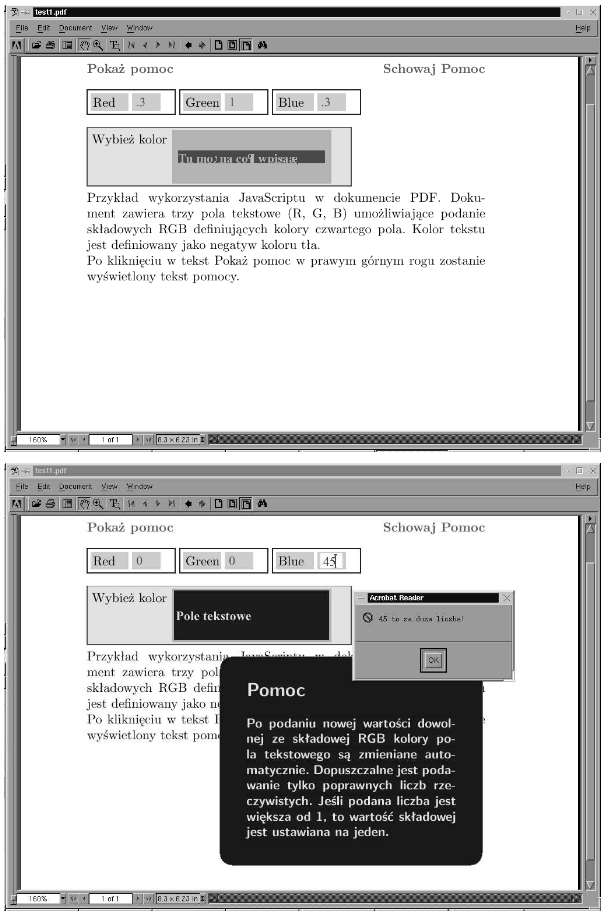
Oczywiście do rozwiązania jest kwestia sposobu definiowania samych formularzy i związywania fragmentów kodu w języku JavaScript ze zdarzeniami generowanymi przez użytkownika. Sprawa nie jest taka prosta jak w przypadku połączenia HTML-a z JavaScript-em. Problem polega na tym, że dokumenty w formacie PDF są praktycznie nieedytowalne; nie mogą być więc tworzone i modyfikowane w edytorach tekstowych. Formularze muszą być tworzone w aplikacji i to zwykle w sposób pośredni: aplikacja tworzy dokument w formacie PostScript zawierający odpowiednie wywołania operatora `pdfmark`, w procesie destylacji (np. programem Acrobat Distiller) powstaje dokument w formacie PDF. Możliwe jest też „ręczne” dodawanie fragmentów kodu do dokumentu PDF w programie Acrobat Exchange, ale jest to skrajnie niewygodne, bo rozdziela proces dołączania skryptu od procesu tworzenia dokumentu – drobna poprawka w tekście i trzeba wszystkie skrypty dołączać jeszcze raz. Ogólnie proces tworzenia dokumentu jest skomplikowany – na pewno przekracza on możliwości nawet dość zaawansowanych i świadomych użytkowników komputerów.

Alternatywą (według mnie najlepszą) jest wykorzystanie możliwości TEX-a. W systemie TEX jest kilka różnych sposobów łatwego tworzenia dokumentów zawierających elementy dynamiczne (formularze) wraz z obsługującym je kodem w języku JavaScript. W szczególności, omówiony wcześniej na łamach Biuletynu system ConTEXt jest najbardziej kompletnym i dopracowanym środowiskiem do tworzenia dokumentów PDF zawierających formularze i JavaScript. Niestety format ten jest jeszcze mało znany, a szkoda, bo jest to moim zdaniem jeden z najciekawszych i najbardziej zaawansowanych systemów do tworzenia bardzo skomplikowanych dokumentów zarówno tradycyjnych (papierowych) jak i elektronicznych. Możliwe jest tworzenie formularzy i korzystanie z JavaScript-u także w LATEX-u – najprościej do tego celu wykorzystać pakiet `hyperref`. Zarówno w przypadku ConTEX-a jak i LATEX-a najlepiej jest użyć pdfTEX-a – zmodyfikowanej wersji TEX-a umożliwiającej bezpośrednie generowanie złożonych dokumentów w formacie PDF.

ConTEXt

Przykład prostego dokumentu utworzonego w systemie ConTEXt pokazano na rys. 1. W tym dokumencie znajdują się trzy pola tekstowe (*Red*, *Green* i *Blue*) przeznaczone do podania wartości składowych RGB koloru, który zostanie użyty jako tło czwartego pola tekstowego („Wybierz kolor” na rysunku). Definicja pól R, G i B wygląda następująco:

```
\setupfield[RGB][label,frame,horizontal]
[ offset=4pt,height=fit,width=95pt,
  framecolor=blue,background=screen ]
[ height=18pt,width=40pt,
  background=screen,
  backgroundscreen=.80,frame=off ]
[ height=18pt,width=30pt,color=red,
  background=screen,
  backgroundscreen=.80,frame=off,
  validate=JS(VALUE),
  afterkey=JS(CHARACTER),
  format=JS(BACKGROUND)     ]

\definefield[R][line][RGB][.5][0]
\definefield[G][line][RGB][.75][0]
\definefield[B][line][RGB][1][0]
...
\hbox{\field[R][Red]
      \field[G][Green]
      \field[B][Blue]}
```



Rys. 1. PDF + JavaScript + TEX = ConTEXt

Jest to typowa dla ConTEXt-a konstrukcja definicyjna – polecenie `\setupfield` definiuje konfigurację (nazwaną tu `RGB`), która może być wykorzystywana przy definiowaniu wielu pól. Całość wygląda przerażająco, ale jest to prosta i typowa konstrukcja definiująca prostokątną ramkę, zawierającą dwie mniejsze prostokątne ramki (etykieta i pole właściwe).

Seria poleceń `\definefield` powoduje zdefiniowanie trzech pól tekstowych (`line`), wykorzystujących podane wcześniej ustawienia (`RGB`). Pierwszym parametrem jest nazwa pola, która może być wykorzytywana w kodzie JavaScript-u do odwoływania się do pól formularza. Ostatnie dwa parametry w definicji pola określają wartości, jakie będą miały pola – wartość początkową i domyślną (po zresetowaniu formularza). W tym przykładowym dokumencie nie korzystamy z wartości domyślnej, ale w definicji pola trzeba je podać.

Ostatnim krokiem koniecznym do utworzenia pola jest umieszczenie go w tekście dokumentu (polecenie `\field`) – tutaj wszytkie trzy pola są umieszczane w jednej linii obok siebie. Pierwszym parametrem jest nazwa pola, a drugim tekst etykiety umieszczony obok pola.

Na końcu części definiującej konfigurację RGB pojawiają się definicje akcji, które mają być wykonane po zajściu różnych zdarzeń związanych z polem:

- zdarzenie `validate` jest generowane po zakończeniu edycji wartości pola – w naszym przypadku sprawdzamy, czy podana wartość jest większa od jedności – jeśli tak, to wyświetlamy ostrzeżenie i ustawiamy wartość pola na 1;
- zdarzenie `afterkey` pojawia się po wpisaniu kolejnego znaku w polu – tutaj chcemy ograniczyć dopuszczalne znaki do cyfr i nie więcej niż jednego znaku kropki dziesiętnej;
- zdarzenie `format` jest generowane po zdarzeniu `validate` – w naszym przykładzie w procedurze jego obsługi odczytujemy wartości pól składowych koloru i zmieniamy wartości kolorów czwartego pola tekstowego.

Kod wykonywany jako obsługa zdarzenia musi być oczywiście także podany w dokumencie. Służy do tego konstrukcja `\startJScode ... \stopJScode`:

```
\startJScode {CHARACTER}
  check_char();
\stopJScode

\startJScode {VALUE}
  check_value();
\stopJScode

\startJScode {BACKGROUND}
  change_bg();
\stopJScode
```

W naszym przykładzie kod obsługujący zdarzenia składa się z wywołania pojedynczych funkcji – nie musi tak być zawsze, ale jest to dobry zwyczaj, bo minimalizuje ilość informacji zapisywanej do wynikowego dokumentu. Jest to ważne zwłaszcza w przypadkach takich jak tutaj, kiedy zdarzenia z kilku pól są obsługiwane przez ten sam kod – jest on zawsze kopiowany w wynikowym dokumencie do miejsca definicji każdego pola. Oczywiście gdzieś muszą być umieszczone definicje funkcji, które będą wywołane gdy zdarzenie zajdzie. Na szczęście jest możliwość podania kodu, który będzie widoczny w całym dokumencie PDF. Do podania takiego kodu – najczęściej właśnie definicji funkcji i globalnych dla dokumentu zmiennych służy konstrukcja `\startJSpreamble ... \stopJSpreamble`. W naszym przypadku są to definicje trzech funkcji:

```
\startJSpreamble {TEST}
function  check_char() {
  if (event.value.indexOf(".") == -1)
    var reDig = /[0-9\.]/;
  else
    var reDig = /[0-9]/;
  if (!reDig.test(event.change))
    event.change="";
}

function check_value () {
  if (eval(event.value) > 1) {
    app.alert(event.value +
              " to za duza liczba!");
    event.value=1;
  } }
function change_bg () {
  var doc=event.target.doc;
  var f=doc.getField("Test");
  var r=doc.getField("R");
  var g=doc.getField("G");
  var b=doc.getField("B");
  f.bgColor=Array("RGB", eval(r.value),
      eval(g.value), eval(b.value));
  f.fgColor=Array("RGB",1-eval(r.value),
      1-eval(g.value),1-eval(b.value));
}
\stopJSpreamble
```

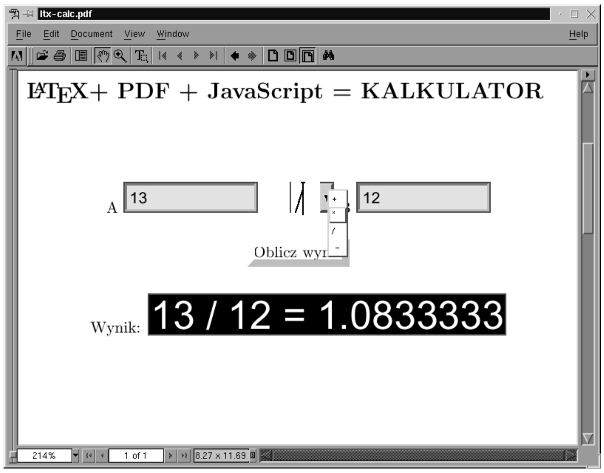
Jak widać można (funkcja change_bg) odwoływać się do pól zdefiniowanych konstrukcjami TEX-owymi.

W przedstawionym przykładzie (na rysunku 1) widać jeszcze jedną ciekawą możliwość jaką daje ConTEXt w połączeniu z formularzami i JAVASCRIPT-em. Jest tu mianowicie krótka pomoc. Tekst pomocy jest umieszczony na przycisku (jak widać przycisk może zawierać nie tylko prosty tekst, ale w zasadzie dowolną kombinację złożonego tekstu i grafiki), który jest nieaktywny (na stałe) i ukryty (na początku) – ma ustawione atrybuty *readonly* i *hidden*. W dokumencie są zdefiniowane dwa hiperodnośniki – jeden powoduje pokazanie, drugi ukrycie tekstu pomocy. Do obsługi tego typu konstrukcji jest wykorzystywany JAVASCRIPT, ale nie odwołujemy się do niego jawnie – wszystko jest ukryte pod eleganckim interfejsem.

```
\setupfields[reset]
\definefield[Help][check]
    [HelpSetup][helptext][helptext]
\setupfield[HelpSetup][width=fit,
           height=fit,frame=off,
           option={readonly,hidden}]
\definesymbol[helptext][\HelpText]

\def\HelpText{%
\framed[frame=on,corner=round,radius=5mm,
       width=10cm,align=normal,
       framecolor=yellow,offset=10mm,
       background=screen,
       backgroundscreen=0.1]{%
\yellow\ss\bf{\tfb\bf Pomoc}\blank[line]
Po podaniu nowej wartości
....
na jeden.\par } }

\setupheadertexts
[{\goto{Pokaż pomoc}[ShowField{Help}]}]
[{\goto{Schowaj Pomoc}[HideField{Help}]}]

\setupfootertexts[]
[\vbox to 0pt{\vss\fitfield[Help]}]
```

LaTeX

W przypadku LaTeX-a możemy wykorzystać udogodnienia pakietu hyperref. Pozwala on definiować formularze oraz umożliwia podawanie kodu JavaScript-owego obsługującego zdarzenia związane z różnymi polami formularzy. Możliwości pakietu hyperref, są ograniczone w porównaniu z tym co oferuje ConTeXt, ale i tutaj można osiągnąć ciekawe efekty. Przykładem niech będzie trywialny kalkulator (rys. 2) – dokument zawierający dwa numeryczne pola tekstowe (do zagwarantowania podania poprawnych liczb użyjemy takiego samego rozwiązania jak poprzednio) oraz listę, z której wybieramy operację, która ma być wykonana na podanych argumentach. Wynik jest wyświetlany w trzecim polu tekstowym. Zmienia się on automatycznie po zmianie wartości w polach tekstowych zawierających argumenty. Ponieważ hyperref, nie pozwala na podanie kodu obsługującego zdarzenia związane z listą wyboru, więc zdefiniowany jest dodatkowy przycisk „Oblicz wynik” wywołujący funkcję obliczającą na żądanie. Pakiet hyperref nie udostępnia także mechanizmów pozwalających podać kod JavaScript globalny dla dokumentu – dlatego używamy tutaj bezpośrednio poleceń niskopoziomowych pdfTeX-a aby dodać dostępne globalnie funkcje, wykorzystywane w różnych częściach dokumentu.



Rys. 2. LaTeX –trywialny kalkulator

```
\documentclass[a4paper]{article}
\usepackage[pdftex,pdfstartview=FitH,
            pdfpagemode=NONE]{hyperref}
\begin{document}

\section*{\LaTeX + {\PDF} + {\JavaScript}
= KALKULATOR}
\begin{center}
\begin{Form}
\TextField[keystroke={check_char()},
  format={calculate()},
  backgroundcolor=1 1 0,borderstyle=I,
  name=op1,width=3cm,height=12pt]{A}
\ChoiceMenu[combo,name=operacja,
  default=+,charsize=20pt,
  height=20pt]{}{+,*,/, - }
\TextField[keystroke={check_char()},
  format={calculate()},
  backgroundcolor=1 1 0,borderstyle=I,
  name=op2,width=3cm,height=12pt]{B}
\bigskip

\PushButton[borderstyle=B,name=policz,
  borderwidth=4,
  backgroundcolor=1 0 0,color=1 1 0,
  bordercolor=1 1 1,height=4cm,width=8cm,
  onclick={calculate()}]
  {\ Oblicz wynik\ }
\bigskip

\TextField[name=wynik,
```

```
  backgroundcolor=0 0 0,
  charsize=25pt,height=25pt,
  color=1 1 1,width=8cm]{Wynik:}
\end{Form}
\end{center}

\pdfobj { << /S /JavaScript /JS (
function  check_char() {
  if (event.value.indexOf(".") == -1)
    var reDig = /[0-9.]/;
  else
    var reDig = /[0-9]/;
  if (!reDig.test(event.change))
    event.change="";
}
function calculate() {
 var doc=event.target.doc;
 var a=doc.getField("op1").value;
 if (!a) a=doc.getField("op1").value=0;
 var b=doc.getField("op2").value;
 if (!b) b=doc.getField("op2").value=0;
 var oper=doc.getField("operacja").value;
 var wynik=doc.getField("wynik");
 wynik.value=a + " " + oper
            + " " + b + " = " +
 String(eval(a + " " + oper + " " + b));
} ) >> }
\pdfobj{<< /Names [ (Document Scripts)
      \the\pdflastobj\space 0 R ] >>}
\pdfnames{/JavaScript
      \the\pdflastobj\space 0 R }
\end{document}
```

Pokazane tutaj przykłady to tylko wierzchołek góry lodowej. Potencjalne zastosowania zależą nie tylko od pomysłowości twórców dokumentów, ale i od znajomości narzędzi i samego formatu PDF – dlatego zainteresowanych odsyłam do dokumentacji.

Literatura

[1] Adobe System Inc.: *Acrobat Forms JavaScript Object Specification – Adobe Technical Note no 5186*, styczeń 1998.

[2] Sebastian Rahtz: *Hypertext marks in LaTeX: the hyperref package*, czerwiec 1998.

[3] Han The Thanh, Sebastian Rahtz, Hans Hagen: *The pdfTeX user manual*, marzec 1999.

[4] Hans Hagen, Ton Otten: *ConTeXt up–to–date 1998/1 Fields Widgets References*, maj 1999.

[5] Adobe Systems Incorporated: *Portable Document Format Reference Manual, Version 1.3*, marzec 1999.

◇ Piotr Bolek
PBolek@GUST.org.pl

Rabin, profesor i karabin maszynowy

Jakub Chabik

Już ponad milion Polaków stale korzysta z Internetu. Patrząc na liczby, trudno jeszcze powiedzieć, by trafił on „pod strzechy", ale można odnieść takie wrażenie, gdy czyta się niektóre wypowiedzi. Wygląda na to, że kulturą wypowiedzi internetowej rządzą zupełnie inne reguły niż wypowiedziami przenoszonymi przez media tradycyjne. Jakie to reguły? Dlaczego tak się dzieje? Jak postępować podczas dyskusji, aby była ona spokojna i konstruktywna? Na te pytania postaram się odpowiedzieć z perspektywy 7-letniego doświadczenia administratora jednej listy dyskusyjnej, a uczestnika wielu.

Kultura elektroniczna

Na początku lat 80., kiedy Internet ostatecznie przestał być siecią wojskową, a nabrał naukowego charakteru, jego twórcy sądzili, iż tworzą medium do wykonywania zdalnych obliczeń, przesyłania ich wyników, łączenia zespołów badaczy. Tymczasem najpopularniejszą usługą stała się poczta elektroniczna. Niedługo potem pojawiły się pierwsze „listserwery" – programy służące do tworzenia tematycznych list dyskusyjnych.

Wraz z pojawieniem się Usenetu internauci zyskali pełne spektrum mediów do prowadzenia dysput. Do Internetu może podłączyć się każdy i każdy może wysłać wypowiedź na grupę dyskusyjną. Jeśli nie jest ona moderowana, to wypowiedź ta zostanie wysłana do wszystkich, którzy chcą ją przeczytać (zapisali się na listę albo czytają grupę). Nie ma ograniczeń czasowych ani geograficznych. Każdy uczestnik takiego elektronicznego forum jest równy, a ewentualne różnice mogą wynikać jedynie z autorytetu, jaki mają dla niego pozostali.

Cechy te wymusiły pewną kulturę, choć nie jestem do końca pewien, czy słowo „kultura" jest tutaj właściwe. W każdym razie internauci porozumiewają się na listach dyskusyjnych w pewien szczególny sposób. Można wyróżnić kilka cech charakterystycznych dyskusji internetowej.

Przede wszystkim większość wypowiedzi na listach dyskusyjnych razi niechlujstwem formy. Błędy ortograficzne, niezręczności stylistyczne, słaba spójność i logika wywodu to niemal reguła. Co gorsza, wszelkie próby zwrócenia uwagi na formę prowokują opór i nieprzychylne komentarze („Purysta językowy się znalazł!").

Kolejna cecha takiej wypowiedzi to duża dawka emocji, czasem przeradzającej się w otwartą agresję. Częściowo jest to wynikiem faktu, iż większość internautów to ludzie młodzi i bardzo młodzi, którzy nie zdążyli nabrać ogłady polemicznej. Ale chyba najważniejsze są przyczyny techniczne. Krótki czas reakcji (wszak internetowe listy trafiają do adresata w ciągu kilku minut), a także oddalenie geograficzne (nie widać reakcji drugiej osoby) sprawiają, że wypowiadamy się śmielej niż podczas zwykłej rozmowy. Jednocześnie Internet daje anonimowość i wiele osób ukrywa się za jej zasłoną. Pozwala ona pleść największe bzdury, szydzić z innych, a nawet miotać obelgi – i nic za to nie grozi. W najgorszym razie zostanie się wykluczonym z grona dyskutantów.

Galeria dyskutantów

Świat dyskusji internetowej zaludniają archetypowe osoby, które pozwoliłem sobie usystematyzować i opisać poniżej.

Expert. Expert (koniecznie przez x!) we własnym przekonaniu zna się na wszystkim. Jeżeli na liście poświęconej podróżom jest mowa o Afryce, to Expert wspomina safari na kenijskiej sawannie. Gdy chodzi o wypady narciarskie – Expert doradzi, jakie „deski" dobrać na alpejskie stoki. Rozmowa schodzi na Włochy – Expert okazuje się niezastąpionym przewodnikiem po kafejkach w okolicach Piazza di Spagna w Rzymie. Można przypuszczać, że poza Internetem Expert sprawdza się doskonale jako prawnik amator, wszystkowiedzący lekarz domowy i osoba chętnie udzielająca porad małżeńskich.

Expert nadaje pewien niepowtarzalny koloryt elektronicznemu forum. Nikt nie bierze zbyt poważnie tego, co ów typ napisał, ale wszyscy czują się pewniej wiedząc, iż nie będą musieli odpowiadać na najbardziej banalne pytania, gdyż zrobi to za nich Expert.

Młody-gniewny. Młody-gniewny wcale nie musi być młody wiekiem, najczęściej jest młody tylko stażem w Internecie. Nie jest wszystkowiedzący, jak Expert, ale za to jego twierdzenia są bardziej kategoryczne. Młodego-gniewnego najłatwiej spotkać na listach poświęconych polityce, ponieważ tutaj – jak wiadomo – emocje są największe.

Młody-gniewny jest niezmordowanym polemistą. Jeżeli dwóch przedstawicieli tego gatunku przyjmie przeciwne stanowiska w dyskusji, to będą obrzucali się błotem tak długo, aż zapomną, czego dotyczył spór. Wtedy na ogół wkroczy Rabin (patrz dalej) i pogodzi ich.

Zagubiony. „Cześć, jestem tu nowy i chciałem zapytać ..." – tak zaczyna się list od Zagubionego. Na liście poświęconej językowi programowania potrafi zapytać o to, jaką rozdzielczość monitora najlepiej ustawić, a na forum, gdzie dyskutowana jest problematyka integracji europejskiej – o ceny dziennego karnetu na paryskie metro. Jest rozbrajający w swojej niewiedzy i naiwności.

Z myślą o Zagubionych tego świata uczestnicy list dyskusyjnych przygotowują tzw. FAQ (zestawy odpowiedzi na najczęściej zadawane pytania). Jest to oczywiście pozbawione sensu, bo Zagubiony o istnieniu takiego dokumentu nie wie, a gdyby nawet wiedział, to i tak przecież łatwiej zapytać, niż czytać kilkanaście stron.

Zagubiony żyje w symbiozie z Expertem i Profesorem. Uwielbiają oni Zagubionych, gdyż dzięki ich pytaniom mogą wykazać się wiedzą.

Statua. Jest ponad to wszystko. Dyskusję prowadzi zwykle z dwoma, trzema wybranymi partnerami, ignorując pozostałych. Poglądy bądź informacje, które są niezgodne z przedstawianymi przez Statuę, po prostu zbywa milczeniem. Przedstawiciel tego gatunku często odpowiada na listy z dużym opóźnieniem. Jest to z jednej strony wynik braku czasu, z drugiej zaś sposób na opadnięcie emocji.

Statua zwykle naprawdę zna się na tej dziedzinie, na której temat się wypowiada. Doskonale potrafi moderować dyskusję i skierować ją na merytoryczne pole. Statua jest bardzo przydatna przy tworzeniu FAQ listy. Jeżeli już coś napisze, to jej wypowiedzi można bez poprawek i redagowania wkleić w taki dokument i ma się pewność, że są one prawdziwe i kompletne.

Karabin Maszynowy. Kiedy łączę się z serwerem pocztowym i widzę komunikat receiving message 1 of ... – i tutaj bardzo duża liczba – wiem, że moją listę dyskusyjną nawiedził Karabin Maszynowy. Dziewięćdziesiąt procent wypowiedzi pisanej przez przedstawiciela tego gatunku to cytat. Pod spodem znajdują się dwa albo trzy zdania dopisane przez autora i jego podpis, najczęściej na kilka linijek. A liczba wypowiedzi wysłanych na jedno forum przekracza wszelkie granice.

Karabin Maszynowy za nic ma gramatykę i ortografię języka polskiego. („Przecież chodzi o to, żeby się rozumieć, a nie o to, żeby wszystko było na piątkę") Ma także tendencję do odpowiadania, zanim dokładnie przeczyta list, na który odpowiada. W efekcie czasami więc polemizuje z poglądami, które powstały wyłącznie w jego umyśle.

Zdarza się, że Karabin Maszynowy jest jednocześnie Młodym-gniewnym. Takie forum staje się zdominowane przez jedną, agresywną osobę i trzeba ogromnego wysiłku (włącznie z wyrzuceniem niewygodnego dyskutanta), aby przywrócić równowagę, bez której nie ma merytorycznej dyskusji.

Profesor. Profesora łatwo poznać po długości listów. Rozmiar jego wypowiedzi często przekracza granice zdrowego rozsądku, a wypowiedzi na jeden ekran Profesor nie pisze nigdy. Każdy z takich listów to mała rozprawa naukowa. Do dziś przechowuję jeden list, gdzie mój adwersarz – przedstawiciel gatunku Profesor – z początku podążył tokiem mojego rozumowania, doprowadził je do absurdu, a potem dosłownie wdeptał mnie w ziemię siłą argumentów. Fakt, że wówczas wykazałem się karygodną niewiedzą, nie sprawia mi przyjemności, ale nadal podziwiam kunszt polemiczny wykazany przez Profesora.

Profesor jest niezwykle cennym nabytkiem na każdej liście dyskusyjnej. W odróżnieniu od Experta, rzeczywiście zna się na dziedzinie, na której temat się wypowiada. Nie ożywia wprawdzie listy, tak jak Karabin Maszynowy, ale warto go czytać, a nawet archiwizować jego wypowiedzi.

Rabin. Jest taki kawał żydowski: Przychodzi trzech kupców do rabina. Icek jest mi winny pieniądze i powinien je oddać – skarży się Mosze. Masz rację – potwierdza rabin. To niesłychane kłamstwo, rabbi, to Mosze jest mi winny pieniądze! – protestuje Icek. Też masz rację – mówi rabin. Ależ rabbi, tylko jeden z nich może mieć rację! – mówi trzeci Żyd. I ty też masz rację! – mówi rabin.

Rabin internetowy próbuje godzić zwaśnionych dyskutantów poprzez akcentowanie najmniej konfliktowych i najbardziej merytorycznych fragmentów wypowiedzi każdego z nich. Lista, która nie ma swojego rabina, albo zamieni się w pobojowisko (wszyscy, poza skonfliktowanymi stronami, uciekną), albo przekształci w listę moderowaną.

Poglądy rabina, czy też jego wiedza, czasami wcale nie są istotne. Unika on dyskusji na tematy merytoryczne – zresztą nikt jej od niego nie oczekuje. Rabin ma uspokajać, a inni zrobią resztę.

Przytulne sadyby elit

Internetowe grupy dyskusyjne są ziszczeniem marzeń oświeceniowych myślicieli o nieskrępowanej wymianie poglądów. Wymiany takiej nie ogranicza przestrzeń geograficzna – nie jest ona bowiem przeszkodą dla mediów elektronicznych. Nie krępują jej autorytety, bo w Internecie Arystoteles i Tomasz Morus, gdyby żyli, byliby równi licealiście z Szamotuł czy sekretarce z Grójca. Wreszcie, nie krępuje jej czas – każdy czyta grupę dyskusyjną wtedy, kiedy ma czas i ochotę, a jego odpowiedź jest natychmiast rozsyłana do wszystkich.

Jest w polskim Internecie lista „Dziennikarz", w zamyśle przeznaczona dla środowiska dziennikarskiego. Nie widziałem tam nigdy wypowiedzi szanowanych dziennikarzy mediów krajowych, choć skądinąd wiem, iż większość z nich z Internetu korzysta regularnie. Pełno tam natomiast samozwańczych analityków od polityki krajowej i zagranicznej, niezmordowanych polemistów, którzy zawsze mają rację, a także propagatorów jakichś idei, które – co za pech! – nie mogą się przebić w mediach tradycyjnych.

Wątpię więc, czy Jan Jakub Rousseau albo Kartezjusz włączyliby się w dzisiejsze internetowe dyskusje. Podejrzewam, że zrobiliby to, co dziś robi większość ludzi, którzy mają coś ciekawego do powiedzenia: wycofaliby się w przytulne grupy moderowane i korespondencję prywatną. Obserwując ten proces, trudno oprzeć się wrażeniu, że po okresie zachłyśnięcia się totalną wolnością wypowiedzi, środowiska internetowe powoli zauważają, iż demokracja i anarchia to jednak dwie różne rzeczy.

Krótki kurs samoobrony

Jak korzystać z różnorodności i obfitości internetowych grup dyskusyjnych, a jednocześnie nie zostać przytłoczonym bezładem i agresją na nich panującymi? Powoli pojawiają się rozwiązania służące do automatycznej selekcji informacji. Najprostszym z nich jest tak zwana kill-lista, czyli lista adresów osób, których wypowiedzi nas nie interesują. Nieco bardziej wysublimowane rozwiązanie proponują serwery, takie jak DejaNews, które pozwalają stworzyć profil osobistych preferencji i „personalizować" usługi.

Najdojrzalszym rozwiązaniem chwalił mi się kiedyś mój internetowy przyjaciel. Otóż, posiada on algorytm przyznający każdej wypowiedzi punkty dodatnie bądź ujemne. List o rozsądnej długości (nie za krótki i nie za długi), dobrze sformatowany, pochodzący od znanego nadawcy albo od osoby podpisującej się imieniem i nazwiskiem, otrzyma punkty dodatnie. List, w którym więcej jest cytatów niż wypowiedzi, sformatowany niechlujnie, pochodzący od anonimowego nadawcy, posiadającego konto na bezpłatnym serwerze, otrzyma punkty ujemne. Mój przyjaciel, w zależności od czasu, jakim dysponuje w danym dniu, określa dolną granicę punktacji, poniżej której listy nie powinny być pokazywane. Potrzeba nieco wiedzy technicznej, aby stworzyć takie rozwiązanie, ale ktoś, kto codziennie ma do przejrzenia kilkaset wypowiedzi, musi sobie jakoś radzić.

Dopóki jednak tego typu rozwiązania nie staną się powszechnie stosowane w narzędziach dostępnych dla każdego, dopóty pozostaje nam polegać na dwóch czynnikach: intuicji i doświadczeniu. Mimo całej tej informatyki wokół nas, warto pamiętać, że to nadal dobrzy doradcy.

Przedruk z Computerworld, nr 11/2000.
Za zgodę dziękujemy autorowi i redakcji CW.