TEXnologia

ConTEXt – pierwsze kroki

Piotr Bolek
Grzegorz Sapijaszko

ConTEXt jest bardzo interesującym nowym formatem TEX-owym. Pozwala łatwo tworzyć skomplikowane dokumenty papierowe i elektroniczne. O sile ConTEXt-a stanowi przemyślany i jednolity interfejs, możliwość łatwego definiowania układu graficznego publikacji niezależnie od jej treści oraz łatwy dostęp z poziomu TEX-a do hipertekstowych funkcji formatu PDF. W tym artykule są przedstawione podstawowe możliwości jakie daje ConTEXt. Nie jest to wyczerpujący opis formatu ale raczej pomoc w rozpoczęciu pracy z tym – naszym zdaniem – niezwykle ciekawym systemem.

Używanie ConTEXt-a

Podobnie jak i w przypadku używania innych formatów, przetwarzanie ConTEXt-em można wywołać poleceniem:

```
pdfetex &cont-en
   --progname=context nasz_plik
```

Najwygodniej jest jednak skorzystać ze standardowego skryptu `texexec`:

```
texexec nasz_plik
```

TEXexec

TEXEXEC został napisany w PERL-u i do jego uruchomienia potrzebny jest działający interpretator tego języka. W przypadku systemów uniksowych nie ma z tym problemów, w pozostałych systemach też da się go zainstalować. Dodatkową zaletą korzystania z TEXEXEC-a jest fakt, iż skrypt automatycznie przetwarza `nasz_plik` wymaganą liczbę razy, co ma szczególne znaczenie przy tworzeniu spisów treści, skorowidzów itp.

Ponadto TEXEXEC pozwala na podanie w wierszu poleceń wielu opcji, które wpływają na wygląd końcowy dokumentu albo sposób jego przetwarzania. Na przykład opcja `--mode=A4` spowoduje wygenerowanie dokumentów o wymiarach strony równych formatowi A4, zaś `--mode=screen` umożliwi przygotowanie (z tego samego pliku źródłowego) dokumentu przeznaczonego do czytania na ekranie monitora. W równie prosty sposób można sterować formatem wyjściowym: po użyciu `--output=pdftex` otrzymamy plik `.pdf`, a po użyciu `--output=dvips` – plik `.dvi`.

Rozszerzmy przedstawioną wyżej składnię uruchamiającą TEXEXEC-a:

```
texexec --mode=A4
   --output=pdftex nasz_plik
```

Jeżeli dokonywaliśmy niewielkich zmian w dokumencie i nie zmieniła się liczba stron lub też nie zależy nam na poprawnych odnośnikach hipertekstowych można „zabezpieczyć się” się przed wielokrotnym uruchamianiem pdfTEX-a:

```
texexec --once nasz_plik
```

Jeżeli pliki przygotowywane były w środowisku MS Windows wygodną opcją może okazać się możliwość przekodowania tekstu „w locie”:

```
texexec --translate=cp1250pl nasz_plik
```

ConTEXt wykorzystuje do tego celu tablice konwersji `*.tcx`. Odpowiednia strona kodowa może być też zapisana w samym dokumencie, np. dla strony ISO-8859-2 pierwszy wiersz dokumentu powinna wyglądać następująco:

```
%& --translate-file=il2-pl
```

TEXutil

Utworzenie spisu treści, ilustracji czy skorowidza wymaga dwukrotnej kompilacji. Ponadto, przy tworzeniu skorowidza dane należy posortować. ConTEXt automatycznie przetwarza dokument wymaganą liczbę razy, tworząc jednocześnie wszystkie niezbędne pliki pomocnicze. W tym celu korzysta on ze skryptu TEXUTIL, którego zadaniem jest przetwarzanie plików pomocniczych, sortowanie lub konwersja.

W normalnej pracy z ConTEXt-em, nie ma potrzeby wykorzystywania TEXUTIL z linii poleceń, gdyż wszystkie konieczne operacje będzie przeprowadzał on sam. Dodatkową zaletą tego skryptu jest możliwość generowania dokumentacji z plików źródłowych ConTEXt-a, np:

```
texutil --documents nazwa_pliku
```

lub też analizowania rysunków zapisanych jako mapy bitowe. Polecenie

```
texutil --figures *.png
```

spowoduje przeszukanie bieżącego katalogu i zapisanie informacji o znalezionych plikach .png w pliku texutil.tuf.

Struktura dokumentu

Podobnie jak w LaTeX-u dokument ConTeXt-owy składa się z dwóch wydzielonych części: definicyjnej, zwanej „set-up area", oraz treści. Poniżej przykład najprostszego dokumentu:

```
%& --translate-file=il2-pl
\setupbodyfont[plr,11pt]
\mainlanguage[pl]
\language[pl]

\starttext
To jest pierwszy dokument w \ConTeXt-ie
z polskimi diakrytykami: ą, ć, ę
\stoptext
```

Wszystko co znajduje się przed \starttext należy do części definicyjnej; tutaj należy przedefiniowywać istniejące bądź definiować nowe środowiska. W powyższym przykładzie \setupbodyfont[plr,11pt] ustawia fonty plr, a \mainlanguage[pl] – język polski, jako domyślne dla całego dokumentu.

Wewnątrz \starttext...\stoptext dokument może być podzielony na cztery składniki:

1. początek
2. środek
3. koniec
4. dodatki

które zdefiniowane są poleceniami

```
\startfrontmatter ... \stopfrontmatter
\startbodymatter  ... \stopbodymatter
\startbackmatter  ... \stopbackmatter
\startappendices  ... \stopappendices
```

Obszary frontmatter i backmatter różnią się od części środkowej m.in. tym, że nagłówki rozdziałów (chapter) nie są numerowane – fakt ten można wykorzystać do tworzenia wszelkiego rodzaju spisów (treści, wykorzystywanych ilustracji czy tabel).

ConTeXt a sprawa polska

Standardowo ConTeXt jest skonfigurowany dla języków: angielskiego, niemieckiego oraz holenderskiego. Przystosowanie go do polskich reguł przenoszenia sprowadza się do usunięcia znaku komentarza w odpowiednim wierszu pliku cont-usr.tex:

```
% \installlanguage [\s!no] [\c!status=\v!start] % norwegian
\installlanguage [\s!pl] [\c!status=\v!start] % polish
% \installlanguage [\s!pt] [\c!status=\v!start] % portuguese
```

oraz wygenerowania nowego formatu. W tym celu używamy TeXexec-a z opcją --make:

```
texexec --language=pl
   --bodyfont=plr --make en
```

Utworzony zostanie format z interfejsem w języku angielskim, jednocześnie domyślnym językiem dla wszystkich dokumentów będzie język polski, zaś dokumenty będą składane czcionkami plr.

Polskie nazwy „elementów typograficznych", np. rozdział, tabela, wiersz, oraz nazwy dni tygodnia, miesięcy zdefiniowane są w pliku lang-sla.tex. Można je również przedefiniować z poziomu dokumentu poleceniem \setupheadtext, np.:

```
\setupheadtext[pl][contents=Spis treści]
```

Podstawowe polecenia ConTeXt-a

W dalszej części na prostych przykładach zaprezentujemy podstawowe polecenia formatu ConTeXt, które potrzebne są w większości dokumentów. Omówione będą polecenia do:

- oznaczania śródtytułów,
- tworzenia list i wyliczeń,
- dołączania ilustracji,
- definiowania nagłówków i stopek oraz sposobu numeracji stron,
- definiowania układu dokumentu

Śródtytuły

Śródtytuły służą do zaznaczenia struktury dokumentu. ConTeXt udostępnia kilka gotowych poleceń służących do tworzenia śródtytułów są to:

- śródtytuły numerowane:
  - \chapter,
  - \section,
  - \subsection,
  - \subsubsection, itd.
- śródtytuły nienumerowane:
  - \title,
  - \subject,
  - \subsubject,

– `\subsubsubject`, itd.

Opcjonalnie możliwe jest podanie etykiety, która będzie wykorzystywana do tworzenia odnośników i odwołań. Oczywiście sposób formatowania śródtytułów można zmieniać używając polecenia `\setuphead`. Można także poleceniem `\definehead` definiować własne śródtytuły i określać ich formatowanie poleceniem `\setuphead`.

Oto przykłady użycia standardowych poleceń:

```
\starttext

\title{Tytuł nienumerowany}
\subject{Tytuł niższego rzędu}
\subsubject{Jeszcze niższego}
\subsubsubject{I tak dalej}

\section{Tytuł numerowany}
\subsection{Tytuł niższego rzędu}
\subsubsection{Jeszcze niższego}

\stoptext
```

Efekt przetworzenia powyższego przykładowego dokumentu jest pokazany z lewej strony rysunku 1.

Przykład definiowania własnych poleceń do oznaczania śródtytułów pokazany jest poniżej:

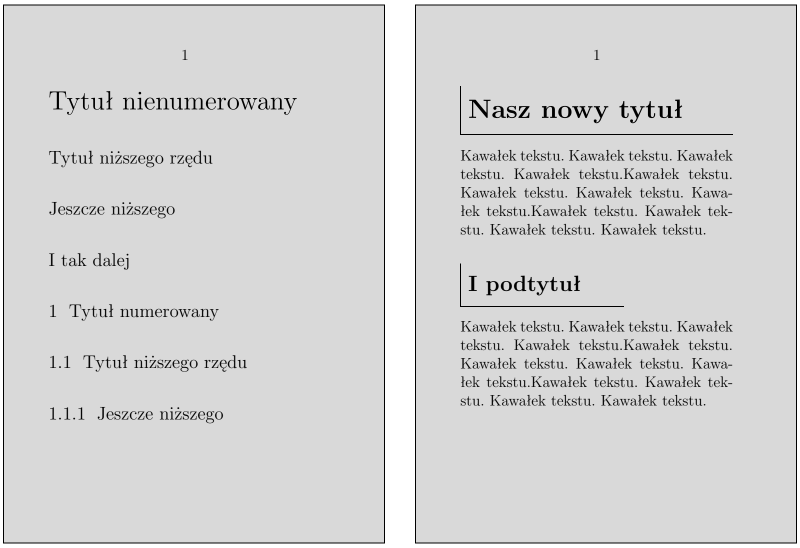
```
1  \definehead[Tytul][section]
2  \definehead[PodTytul][subsection]

3  \setuphead[Tytul]
4    [style={\bfc},
5     textcommand=\makehead{\textwidth}]

6  \setuphead[PodTytul]
7    [style={\bfb},
8     textcommand=\makehead{.6\textwidth}]

9  \setuphead[Tytul,PodTytul]
10   [number=no]

11 \def\makehead#1#2{\framed
12     [ offset=2mm,
13       frame=off,
14       bottomframe=on,
15       leftframe=on,
16       width=#1,
17       align=right ]{#2}}

18 \def\tekst{Kawałek tekstu. Kawałek tekstu.
19 Kawałek tekstu. Kawałek tekstu.}

20 \starttext
21 \Tytul{Nasz nowy tytuł}
22 \tekst\tekst\tekst

23 \PodTytul{I podtytuł}
24 \tekst\tekst\tekst
25 \stoptext
```

Definiujemy tutaj (wiersze 1 i 2) dwa nowe polecenia: `\Tytul` i `\PodTytul`. Polecenie `\definehead`, używane do definiowania nowych rodzajów śródtytułów, wymaga podania dwóch parametrów: pierwszym jest nazwa nowego śródtytułu (bez znaku w-tył-ciach), a drugim nazwa istniejącego śródtytułu, z którego będzie wzięty domyślny sposób formatowania. Następnie używając polecenia `\setuphead` (2–10) zmieniamy domyślny sposób formatowania. Jak widać możliwe jest definiowanie pewnych parametrów jednocześnie dla więcej niż jednego rodzaju śródtytułu, jeśli chcemy ustawić je tak samo (9–10). Opcja `textcommad` służy do podania polecenia jednoargumentowego, albo – tak jak tutaj – polecenia wieloargumentowego z podanymi wartościami dla wszystkich oprócz ostatniego argumentu. Polecenie to zostanie wywołane w miejscu wystąpienia śródtytułu, z tekstem jako wartością ostatniego niepodanego w definicji argumentu.

Do formatowania obu śródtytułów używamy tego samego dwuargumentowego polecenia `\makehead` wywoływanego z inną wartością pierwszego argumentu. W poleceniu tym używamy instrukcji `\framed` – jest to często wykorzystywana konstrukcja (w wielu przypadkach pośrednio), która umożliwia definiowanie prostokątnych obszarów. Dla takich obszarów możemy podać różne parametry: tło (nie tylko kolor ale, jeśli wykorzystamy mechanizm nakładek – overlays, także grafikę), kolor i kształt ramki oraz wiele innych. Znaczenie parametrów wykorzystywanych w przykładzie jest chyba oczywiste zwłaszcza po obejrzeniu efektu sformatowania naszego dokumentu po prawej stronie rysunku 1.

Cenną właściwością poleceń służących do oznaczania śródtytułów jest możliwość rozpoczęcia używania od dowolnego poziomu zagłębienia bez znanego użytkownikom LaTeX-a efektu zaczynania numeracji od zera. Np. poniższy śródtytuł otrzyma numer „1”, a nie „0.1” czy „0.0.1”.:

```
\starttext
\subsection{Od środka?}
\stoptext
```



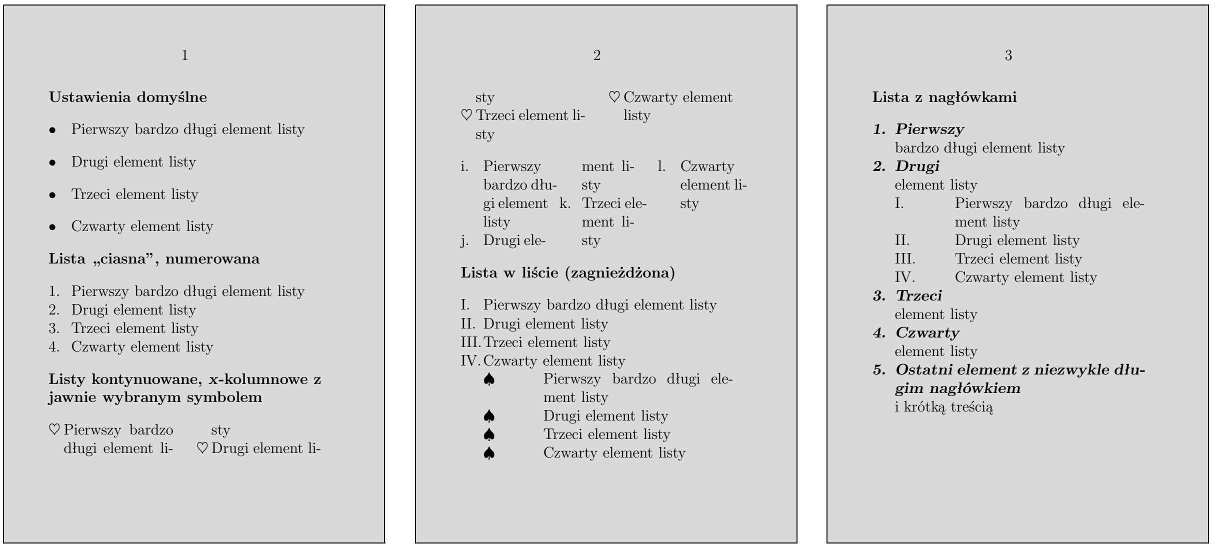
Rysunek 1: Śródtytuły

Listy i wyliczenia

Listy i wyliczenia tworzy się używając środowiska[1] `itemize`. Do zadawania globalnych parametrów wykorzystywanych przy formatowaniu listy służy polecenie `\setupitemize`. Kolejne elementy listy oznacza się używając polecenia `\item` albo, jeśli chcemy żeby elementy listy zaczynały się od nagłówka, polecenia `\head`. Oto przykłady tworzenia różnego rodzaju list (rys. 2).

```
%
\setupitemize[headstyle=boldslanted]
\setupitemize[2][symbol=S,width=4em]
\definesymbol[H][$\heartsuit$]
\definesymbol[S][$\spadesuit$]

\starttext
\def\items{
\item Pierwszy bardzo długi element listy
\item Drugi element listy
\item Trzeci element listy
\item Czwarty element listy}

{\bf Ustawienia domyślne}
\startitemize
\items
\stopitemize

{\bf Lista ,,ciasna'', numerowana}
\startitemize[n,packed]
\items
\stopitemize

{\bf Listy kontynuowane,
{\em x}-kolumnowe z jawnie
        wybranym symbolem}
\startitemize
[continue,H,packed,columns,serried,two]
\items
\stopitemize

\startitemize
[continue,a,columns,three]
\items
\stopitemize

{\bf Lista w liście (zagnieżdżona)}
\startitemize[R,packed]
\items
\startitemize[packed]
\items
\stopitemize
\stopitemize

{\bf Lista z nagłówkami}
\startitemize[n,packed]
\head Pierwszy\par bardzo długi element
listy
\head Drugi\par element listy
\startitemize[R,packed]
\items
\stopitemize
```

[1] Parę poleceń `\start...` – `\stop...` będziemy przez analogię do znanej konstrukcji LaTeX-owej nazywać środowiskiem



Rysunek 2: Listy i wyliczenia

```
45 \head Trzeci\par element listy
46 \head Czwarty\par element listy
47 \head Ostatni element z niezwykle
48 długim nagłówkiem\par
49 i krótką treścią
50 \stopitemize
51 \stoptext
```

Symbole, które będą używane do zaznaczenia elementów listy można podać jako opcję na początku listy, albo w części definicyjnej. Jeśli nie podamy symbolu, to otrzymamy listę nienumerowaną. W kolejnych poziomach zagłębienia listy do zaznaczenia elementów będzie używany inny symbol. Możliwe jest też definiowanie własnych symboli (wiersze 3 i 4) i używanie ich w wyliczeniach (2, 22). Różnego rodzaju listy numerowane uzyskujemy podając w opcjach następujące symbole: `n` – cyfry arabskie, `a` – litery małe, `A` – litery wielkie, `r` – cyfry rzymskie małe, `R` cyfry rzymskie wielkie.

Możemy też definiować globalny sposób formatowania list: w wierszu 1 określamy formatowanie nagłówków wszystkich list (oczywiście tych, w których użyjemy polecenia `\head`). Nagłówkiem jest cały akapit – możemy użyć, tak jak tutaj, polecenia `\par` lub oddzielić nagłówek od tekstu pustym wierszem.

W wierszu 2 określamy sposób formatowania list na drugim poziomie zagłębienia. Zwróćmy uwagę, że ustawienie marginesu dla drugiego poziomu (wiersz 2) jest aktywne także jeśli zmienimy sposób oznaczania elementów. Domyślnie sąsiednie elementy listy są rozdzielone światłem – podanie parametru `packed` powoduje skład bez dodatkowych odstępów.

Listy można także składać w kilku kolumnach – służy do tego opcja `columns` – liczbę kolumn podajemy słownie (tutaj po angielsku).

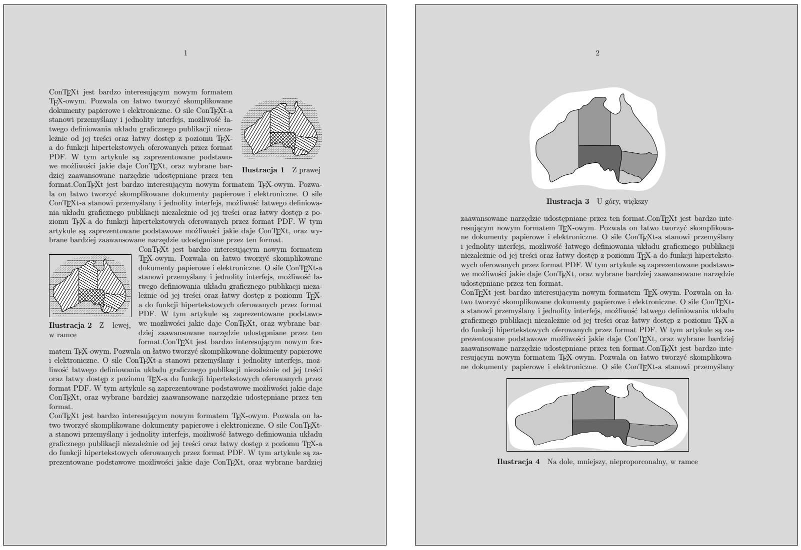
Parametr `continued` powoduje kontynuację numeracji – zauważmy, że numeracja jest kontynuowana także dla list nienumerowanych oraz po zmianie sposobu numeracji (ww. 23, 27).

Ilustracje

Ilustracje można umieszczać używając polecenia `\placefigure`. Polecenie `\placefigure` pełni podobną funkcję jak LaTeX-owe środowisko `figure`. Rezerwuje ono miejsce i pozwala na podanie parametrów związanych z położeniem. Jest ono reprezentantem grupy poleceń umożliwiających definiowanie obiektów przemieszczalnych. Polecenie to ma dwa opcjonalne i dwa wymagane parametry:

```
\placefigure [ <położenie> ][ <etykiety> ]
        { <podpis> }{ <zawartość> }
```

Parametr `położenie` umożliwia podanie preferencji co do miejsca umieszczenia ilustracji. Typowe wartości to: `left`, `right`, `here`, `top`, `bottom`, `page`, `opposite`, `force`. Etykiety pozwalają odwoływać się do ilustracji w sposób symboliczny. Zawartość może być oczywiście dowolna – wartością tego parametru nie musi być polecenie dołączenia zewnętrznego



1

ConTEXt jest bardzo interesującym nowym formatem TEX-owym. Pozwala on łatwo tworzyć skomplikowane dokumenty papierowe i elektroniczne. O sile ConTEXt-a stanowi przemyślany i jednolity interfejs, możliwość łatwego definiowania układu graficznego publikacji niezależnie od jej treści oraz łatwy dostęp z poziomu TEX-a do funkcji hipertekstowych oferowanych przez format PDF. W tym artykule są zaprezentowane podstawowe możliwości jakie daje ConTEXt, oraz wybrane bardziej zaawansowane narzędzie udostępniane przez ten format.ConTEXt jest bardzo interesującym nowym formatem TEX-owym. Pozwala on łatwo tworzyć skomplikowane dokumenty papierowe i elektroniczne. O sile ConTEXt-a stanowi przemyślany i jednolity interfejs, możliwość łatwego definiowania układu graficznego publikacji niezależnie od jej treści oraz łatwy dostęp z poziomu TEX-a do funkcji hipertekstowych oferowanych przez format PDF. W tym artykule są zaprezentowane podstawowe możliwości jakie daje ConTEXt, oraz wybrane bardziej zaawansowane narzędzie udostępniane przez ten format.

Ilustracja 1 Z prawej

ConTEXt jest bardzo interesującym nowym formatem TEX-owym. Pozwala on łatwo tworzyć skomplikowane dokumenty papierowe i elektroniczne. O sile ConTEXt-a stanowi przemyślany i jednolity interfejs, możliwość łatwego definiowania układu graficznego publikacji niezależnie od jej treści oraz łatwy dostęp z poziomu TEX-a do funkcji hipertekstowych oferowanych przez format PDF. W tym artykule są zaprezentowane podstawowe możliwości jakie daje ConTEXt, oraz wybrane bardziej zaawansowane narzędzie udostępniane przez ten format.ConTEXt jest bardzo interesującym nowym formatem TEX-owym. Pozwala on łatwo tworzyć skomplikowane dokumenty papierowe i elektroniczne. O sile ConTEXt-a stanowi przemyślany i jednolity interfejs, możliwość łatwego definiowania układu graficznego publikacji niezależnie od jej treści oraz łatwy dostęp z poziomu TEX-a do funkcji hipertekstowych oferowanych przez format PDF. W tym artykule są zaprezentowane podstawowe możliwości jakie daje ConTEXt, oraz wybrane bardziej zaawansowane narzędzie udostępniane przez ten format.

Ilustracja 2 Z lewej, w ramce

ConTEXt jest bardzo interesującym nowym formatem TEX-owym. Pozwala on łatwo tworzyć skomplikowane dokumenty papierowe i elektroniczne. O sile ConTEXt-a stanowi przemyślany i jednolity interfejs, możliwość łatwego definiowania układu graficznego publikacji niezależnie od jej treści oraz łatwy dostęp z poziomu TEX-a do funkcji hipertekstowych oferowanych przez format PDF. W tym artykule są zaprezentowane podstawowe możliwości jakie daje ConTEXt, oraz wybrane bardziej

2

Ilustracja 3 U góry, większy

zaawansowane narzędzie udostępniane przez ten format.ConTEXt jest bardzo interesującym nowym formatem TEX-owym. Pozwala on łatwo tworzyć skomplikowane dokumenty papierowe i elektroniczne. O sile ConTEXt-a stanowi przemyślany i jednolity interfejs, możliwość łatwego definiowania układu graficznego publikacji niezależnie od jej treści oraz łatwy dostęp z poziomu TEX-a do funkcji hipertekstowych oferowanych przez format PDF. W tym artykule są zaprezentowane podstawowe możliwości jakie daje ConTEXt, oraz wybrane bardziej zaawansowane narzędzie udostępniane przez ten format.

ConTEXt jest bardzo interesującym nowym formatem TEX-owym. Pozwala on łatwo tworzyć skomplikowane dokumenty papierowe i elektroniczne. O sile ConTEXt-a stanowi przemyślany i jednolity interfejs, możliwość łatwego definiowania układu graficznego publikacji niezależnie od jej treści oraz łatwy dostęp z poziomu TEX-a do funkcji hipertekstowych oferowanych przez format PDF. W tym artykule są zaprezentowane podstawowe możliwości jakie daje ConTEXt, oraz wybrane bardziej zaawansowane narzędzie udostępniane przez ten format.ConTEXt jest bardzo interesującym nowym formatem TEX-owym. Pozwala on łatwo tworzyć skomplikowane dokumenty papierowe i elektroniczne. O sile ConTEXt-a stanowi przemyślany

Ilustracja 4 Na dole, mniejszy, nieproporconalny, w ramce

Rysunek 3: Definiowanie i umieszczanie ilustracji

pliku graficznego. Ilustracja 4 jest na przykład zdefiniowana następująco:

```
\placefigure[force][fig:test]
  {Przykładowa ilustracja}
  {\framed[offset=3mm]{\bfc Ramka}}
```

Ramka

Rysunek 4: Przykładowa ilustracja

Niepodanie podpisu nie sprawia, że nasz rysunek nie będzie podpisany – numer zostanie. Jeśli chcemy umieścić ilustrację całkiem bez podpisu powinniśmy jako podpis podać none.

```
\placefigure[force]{none}
  {\framed[offset=3mm]{\bfc Bez podpisu}}
```

Bez podpisu

Grafikę można dołączać z zewnętrznych plików graficznych w różnych formatach zależnych od generowanego formatu wyjściowego (PDF, czy DVI). Przy generowaniu formatu PDF można dołączać grafikę w formatach METAPOST, PDF, TIF, PNG, JPG; w przypadku generowania Postscripta za pośrednictwem formatu DVI, z wykorzystaniem programu dvips, ograniczeni jesteśmy do formatu EPS (w tym także METAPOST).

ConTEXt umożliwia wielokrotne używanie tego samego rysunku. Dlatego aby dołączyć zewnętrzny plik graficzny najpierw definiujemy go używając polecenia \useexternalfigure:

```
\useexternalfigure
  [<nazwa symboliczna>]
  [<nazwa pliku>]
  [<parametry>]
```

Nazwa symboliczna określa nazwę jakiej będziemy później używać do odwoływania się do danego pliku graficznego. Używając parametrów możemy rysunek przeskalować albo jawnie określić format pliku bądź umieścić rysunek w ramce. Zdefiniowany wcześniej rysunek można umieścić poleceniem \externalfigure. Pierwszym parametrem polecenia \externalfigure jest nazwa symboliczna (zdefiniowana poleceniem \useexternalfigure) albo nazwa pliku, drugim parametry (takie same jak dla polecenia \useexternalfigure). Kilka przykładów dołączania obrazków jest pokazanych poniżej, efekt przetworzenia na rys. 3.

```
\starttext

\useexternalfigure
  [australia][a.1]
  [type=mps,width=.3\textwidth]

\placefigure[right]{Z prawej}
{\externalfigure[australia]}
\tekst
```

```
8 \placefigure[left]{Z lewej, w ramce}
9 {\externalfigure[australia][frame=on]}
10 \tekst

11 \placefigure[top]{U góry, większy}
12 {\externalfigure[a.2]
13         [width=.5\textwidth]}
14 \tekst
15 \placefigure[bottom]{Na dole, mniejszy,
16 nieproporconalny, w ramce}
17 {\externalfigure[a.2]
18   [frame=on,height=4cm,width=10cm]}

19 \tekst
20 \stoptext
```

Powyższy przykład jest prezentacją użycia zarówno pary \useexternalfigure – \externalfigure (wiersze 2–4, 6, 9) jak i samego \externalfigure (wiersze 12–13, 17–18).

Numerowanie stron, żywa pagina

Domyślny układ dokumentu jest dosyć nietypowy. Numery stron są umieszczane na środku nagłówka strony. W większości przypadków na pewno będziemy chcieli go zmienić. Można to zrobić używając polecenia \setuppagenumbering, które pozwala określić, położenie i sposób formatowania numeru strony. Najważniejsze opcje i ich wartości to:

alternative
: singlesided, doublesided

location
: header, footer, left, right, middle, inleft, inright, margin, inmargin

conversion
: numbers, characters, Characters, romannumerals, Romannumerals

Opcja conversion określa sposób numeracji – znaczenie kolejnych wartości jest oczywiste – odpowiednio: liczby arabskie, litery małe, litery wielkie, liczby rzymskie pisane małymi i wielkimi literami. Dodatkowo możliwe jest podanie tekstu, który będzie umieszczony na lewo i na prawo od numeru strony (opcje left i right). Oto przykłady

```
\setuppagenumbering[
 conversion=Romannumerals,
 left={---~},
 right={~---},
 location={footer,left},
 alternative=doublesided]
```

Powyższy przykład spowoduje umieszczanie numerów stron z lewej strony stopki na stronach nieparzystych i z prawej strony na stronach parzystych. Numer strony będzie miał postać: „— XII —”.

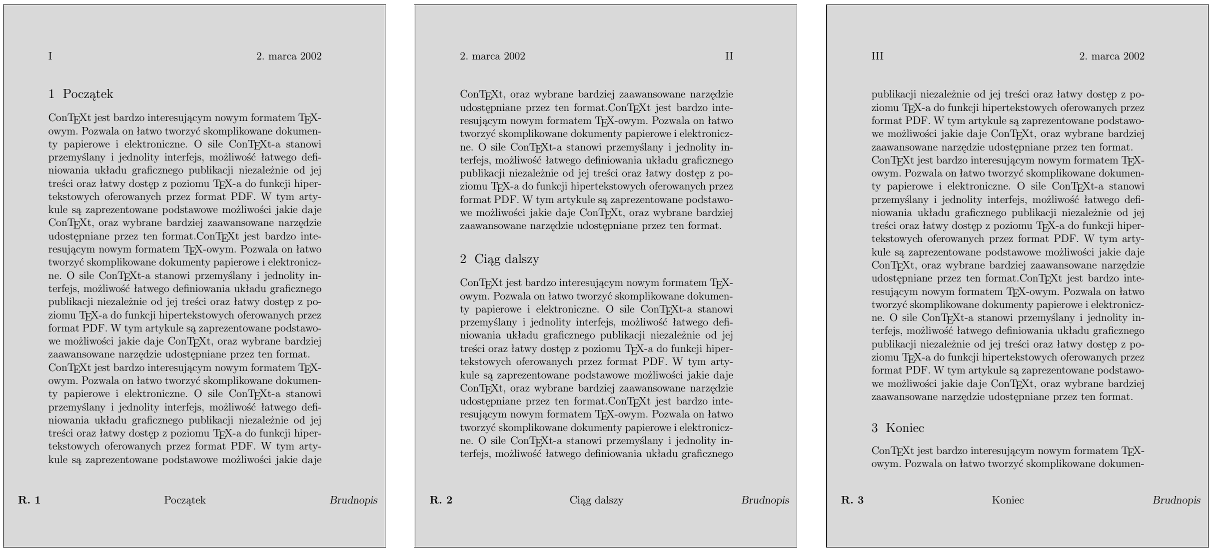
Możliwe jest też sformatowanie numeru strony w dowolny sposób podając w opcji command jednoargumentowe polecenie, które będzie wykorzystywane do formatowania każdego numeru strony. Poniżej na przykład definiujemy numerację typu: „Strona x z y”

```
\setuppagenumbering[command=\Pagenum]
\def\Pagenum#1{Strona #1 z \lastpage}
```

Innym sposobem określenia położenia numeru strony jest użycie poleceń do definiowania zawartości nagłówków i stopek: \setupheadertexts, \setupfotertexts, \setupheader, \setupfooter. Omówimy sposób definiowania zawartości nagłówków, w analogiczny sposób można definiować także stopki.

Poleceniu \setup...texts można podać od jednego do pięciu parametrów. Podając jeden parametr określamy zawartość środkowej części nagłówka, dwa parametry określają zawartość lewej i prawej strony nagłówka, cztery parametry określają zawartość wewnętrznych i zewnętrznych części nagłówka w składzie dwustronnym (nieparzysty wewnętrzny, nieparzysty zewnętrzny, parzysty wewnętrzny, parzysty zewnętrzny). Przy podaniu trzech parametrów pierwszy ma inne znaczenie: określa obszar, którego nagłówek definiujemy: tekst (text), margines (margin) albo skraj (edge).

```
\setuppagenumbering
  [conversion=Romannumerals]
\setupheadertexts[]
\setupheadertexts
  [text][pagenumber][date]
        [pagenumber][date]
\setupfootertexts[section]
\setupfootertexts
  [margin][sectionnumber][{\em Brudnopis}]
\setupfooter
  [margin][leftstyle=\Rozdzial]
\def\Rozdzial{\bf R. }

\starttext
\section{Początek}
\tekst\par\tekst\par
```



I 2. marca 2002

1 Początek

ConTEXt jest bardzo interesującym nowym formatem TEX-owym. Pozwala on łatwo tworzyć skomplikowane dokumenty papierowe i elektroniczne. O sile ConTEXt-a stanowi przemyślany i jednolity interfejs, możliwość łatwego definiowania układu graficznego publikacji niezależnie od jej treści oraz łatwy dostęp z poziomu TEX-a do funkcji hipertekstowych oferowanych przez format PDF. W tym artykule są zaprezentowane podstawowe możliwości jakie daje ConTEXt, oraz wybrane bardziej zaawansowane narzędzie udostępniane przez ten format.ConTEXt jest bardzo interesującym nowym formatem TEX-owym. Pozwala on łatwo tworzyć skomplikowane dokumenty papierowe i elektroniczne. O sile ConTEXt-a stanowi przemyślany i jednolity interfejs, możliwość łatwego definiowania układu graficznego publikacji niezależnie od jej treści oraz łatwy dostęp z poziomu TEX-a do funkcji hipertekstowych oferowanych przez format PDF. W tym artykule są zaprezentowane podstawowe możliwości jakie daje ConTEXt, oraz wybrane bardziej zaawansowane narzędzie udostępniane przez ten format.
ConTEXt jest bardzo interesującym nowym formatem TEX-owym. Pozwala on łatwo tworzyć skomplikowane dokumenty papierowe i elektroniczne. O sile ConTEXt-a stanowi przemyślany i jednolity interfejs, możliwość łatwego definiowania układu graficznego publikacji niezależnie od jej treści oraz łatwy dostęp z poziomu TEX-a do funkcji hipertekstowych oferowanych przez format PDF. W tym artykule są zaprezentowane podstawowe możliwości jakie daje

R. 1 Początek *Brudnopis*

2. marca 2002 II

ConTEXt, oraz wybrane bardziej zaawansowane narzędzie udostępniane przez ten format.ConTEXt jest bardzo interesującym nowym formatem TEX-owym. Pozwala on łatwo tworzyć skomplikowane dokumenty papierowe i elektroniczne. O sile ConTEXt-a stanowi przemyślany i jednolity interfejs, możliwość łatwego definiowania układu graficznego publikacji niezależnie od jej treści oraz łatwy dostęp z poziomu TEX-a do funkcji hipertekstowych oferowanych przez format PDF. W tym artykule są zaprezentowane podstawowe możliwości jakie daje ConTEXt, oraz wybrane bardziej zaawansowane narzędzie udostępniane przez ten format.

2 Ciąg dalszy

ConTEXt jest bardzo interesującym nowym formatem TEX-owym. Pozwala on łatwo tworzyć skomplikowane dokumenty papierowe i elektroniczne. O sile ConTEXt-a stanowi przemyślany i jednolity interfejs, możliwość łatwego definiowania układu graficznego publikacji niezależnie od jej treści oraz łatwy dostęp z poziomu TEX-a do funkcji hipertekstowych oferowanych przez format PDF. W tym artykule są zaprezentowane podstawowe możliwości jakie daje ConTEXt, oraz wybrane bardziej zaawansowane narzędzie udostępniane przez ten format.ConTEXt jest bardzo interesującym nowym formatem TEX-owym. Pozwala on łatwo tworzyć skomplikowane dokumenty papierowe i elektroniczne. O sile ConTEXt-a stanowi przemyślany i jednolity interfejs, możliwość łatwego definiowania układu graficznego

R. 2 Ciąg dalszy *Brudnopis*

III 2. marca 2002

publikacji niezależnie od jej treści oraz łatwy dostęp z poziomu TEX-a do funkcji hipertekstowych oferowanych przez format PDF. W tym artykule są zaprezentowane podstawowe możliwości jakie daje ConTEXt, oraz wybrane bardziej zaawansowane narzędzie udostępniane przez ten format.
ConTEXt jest bardzo interesującym nowym formatem TEX-owym. Pozwala on łatwo tworzyć skomplikowane dokumenty papierowe i elektroniczne. O sile ConTEXt-a stanowi przemyślany i jednolity interfejs, możliwość łatwego definiowania układu graficznego publikacji niezależnie od jej treści oraz łatwy dostęp z poziomu TEX-a do funkcji hipertekstowych oferowanych przez format PDF. W tym artykule są zaprezentowane podstawowe możliwości jakie daje ConTEXt, oraz wybrane bardziej zaawansowane narzędzie udostępniane przez ten format.ConTEXt jest bardzo interesującym nowym formatem TEX-owym. Pozwala on łatwo tworzyć skomplikowane dokumenty papierowe i elektroniczne. O sile ConTEXt-a stanowi przemyślany i jednolity interfejs, możliwość łatwego definiowania układu graficznego publikacji niezależnie od jej treści oraz łatwy dostęp z poziomu TEX-a do funkcji hipertekstowych oferowanych przez format PDF. W tym artykule są zaprezentowane podstawowe możliwości jakie daje ConTEXt, oraz wybrane bardziej zaawansowane narzędzie udostępniane przez ten format.

3 Koniec

ConTEXt jest bardzo interesującym nowym formatem TEX-owym. Pozwala on łatwo tworzyć skomplikowane dokumen-

R. 3 Koniec *Brudnopis*

Rysunek 5: Żywa pagina

```
16 \section{Ciąg dalszy}
17 \tekst\par\tekst\par
18 \section{Koniec}
19 \tekst\par
20 \stoptext
```

Na początku powyższego przykładu (wiersze 1–2) definiujemy sposób numerowania (wielkie cyfry rzymskie). Wiersz trzeci to usunięcie standardowego numeru strony na środku nagłówka. Wiersze 4, 5 i 6, to definicja nagłówka – na zewnętrznej krawędzi strony umieszczamy datę przetworzenia, na wewnętrznej – numer strony (oczywiście zgodnie z początkową deklaracją pisany cyframi rzymskimi). Jawnie określamy tutaj, dla którego obszaru definiujemy nagłówek – w tym przypadku nie jest to właściwie konieczne, ponieważ podając poleceniu `\setup...texts` parzystą liczbę argumentów definiujemy nagłówek i stopkę właśnie dla obszaru `text`. W poleceniu podanym w wierszach 4, 5 i 6 można by więc pominąć pierwszy parametr:

```
\setupheadertexts
  [pagenumber][date]
  [pagenumber][date]
```

Stopkę definiujemy w dwóch krokach – najpierw środek – tam jest umieszczony tekst aktualnego śródtytułu (wiersz 7). Pozostałe elementy stopki umieszczane są na marginesach: na lewym marginesie numer rozdziału (śródtytułu), po prawej złożone kursywą słowo „Brudnopis” (ww. 8–9). Dodatkowo sposób sformatowania numeru rozdziału definiujemy z wykorzystaniem opcji `command` (ww. 10–12). Pierwsze trzy strony z naszego przykładowego dokumentu możemy zobaczyć na rys. 5.

Wymiary papieru i układ strony

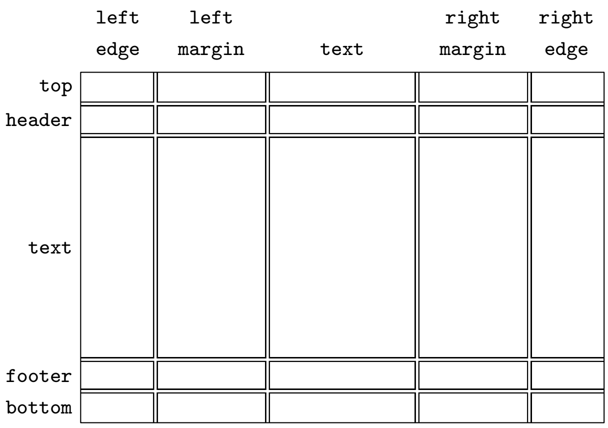
Do podania wymiarów kartki i wielkości strony służy polecenie `\setuppapersize`. Można tu podać jeden lub dwa argumenty – pierwszy z nich określa wymiary pola składu, drugi wymiary papieru, na którym będzie drukowany dokument. Dopuszczalne są następujące wartości parametrów określające typowe wymiary papieru: A0 – A9, B0 – B9, C0 – C9, kilka wymiarów ekranowych (S3 – S6), CD, amerykańskie wymiary papieru, wymiary typowych kopert. Możliwe jest także używanie innych wymiarów zdefiniowanych przez użytkownika poleceniem `\definepapersize`.

Pokazane poniżej polecenie oznacza, że będziemy składać w formacie B5 na papierze A4.

```
\setuppapersize[B5][A4]
```

Strona, na której składamy tekst jest podzielona na wiele obszarów. Jest ich w sumie aż 25 (patrz rys. 6). Ich wymiary i wzajemne położenie określa się poleceniem `\setuplayout`.

Zasadniczy tekst dokumentu jest umieszczany w obszarze „adresowanym” `text, text`. Inne obszary przeznaczone są do umieszczania elementów stałych (lub zmiennych ze strony na stronę): żywa pagina, elementy nawigacyjne, logo itp. W typo-



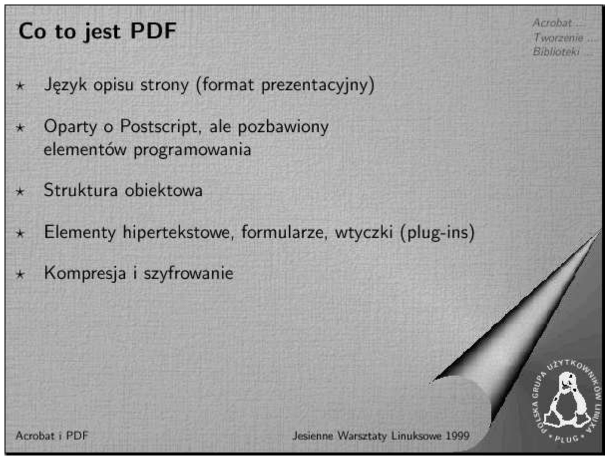
Rysunek 6: Układ strony

wych dokumentach zwykle nie są wykorzystywane wszystkie obszary – najczęściej w przypadku tradycyjnych papierowych dokumentów oprócz obszaru przeznaczonego na zasadniczy tekst dokumentu wykorzystywany jest nagłówek, stopka i marginesy. Pozostałe pola są wykorzystywane zwykle przy tworzeniu interaktywnych dokumentów elektronicznych, umieszcza się w nich elementy nawigacyjne.

Polecenie `\setuplayout` pozwala zdefiniować wielkości obszarów i odstępy między nimi. Przykład definicji układu strony dla dokumentu – prezentacji elektronicznej w formacie PDF.

```
1  \setuppapersize[S6]
2  \setuplayout[
3    rightedge=70pt,
4    leftedge=0pt,
5    edgedistance=15pt,
6    margin=0pt,
7    topspace=10pt,
8    backspace=10pt,
9    height=fit,
10   width=fit,
11   header=0pt,
12   footerdistance=10pt,
13   footer=.75\bodyfontsize]
```

Definiujemy tutaj prosty układ strony, bez marginesów (w. 6), za to z prawym brzegiem o szerokości 70pt (w. 3), odległym od pola tekstowego o 15pt (w. 5). Odstępy od górnej i lewej krawędzi kartki (a właściwie ekranu) są ustawione na 10pt (ww. 7–8). Stopka ma wysokość 75% podstawowego stopnia pisma i jest umieszczona 10pt poniżej pola tekstowego (ww. 12–13). Szerokość i wysokość tekstu jest wyliczana automatycznie (ww. 9–10). Przykładowa strona dokumentu z tak zdefiniowanym układem pokazana jest na rys. 7.



Rysunek 7: Przykładowy dokument w układzie „ekranowym"

Na przykładowym rysunku widzimy zastosowanie tła. Jako tło każdej strony dokumentu użyty jest oczywiście zewnętrzny plik (w tym przypadku była to mapa bitowa przygotowana z wykorzystaniem programu gimp). Omówienie sposobu definiowania tła, nakładek i prostokątów oraz sposobu tworzenia dokumentów interaktywnych i prezentacyjnych pozostawimy na inną okazję.

* * *

Autorem ConTEXt-a jest Hans Hagen. System jest dostępny z archiwum CTAN lub ze strony firmy H. Hagena – Pragma ADE (`www.pragma.ade.nl`). Licencja zezwala na swobodną dystrybucję i wykorzystanie systemu do celów niekomercyjnych. Wykorzystanie do celów komercyjnych wymaga osobnej licencji. Pewnym problemem dla entuzjastów ConTEXt-a jest także brak kompletnej, chociażby anglojęzycznej, dokumentacji.

Pokazane w artykule przykłady i opisane mechanizmy to prezentacja drobnego fragmentu możliwości jakie daje ConTEXt. Mamy nadzieję, że nasz artykuł zachęci czytelników Biuletynu do zainteresowania się tym narzędziem.

Źródła przykładów zaprezentowanych w tym artykule są dostępne na serwerze GUST-u:
`ftp://ftp.gust.org.pl/TeX/GUST/bulletin/14`

◇ Piotr Bolek
Grzegorz Sapijaszko