The InTeX toolkit

G. A. Oliveira

January 23, 2017

1 Introduction

This is the user-manual for InTeX, a typesetting system for interactive-document writing.

2 Overview

Figure 1 shows the steps to generate an interactive document using InTeX. They are:

1. First of all we work in our TeX document and compile it to obtain a PDF file.
    1.1. Import `intex.sty` package to create a new class environment and use its definitions to insert special content into the document .
    1.2. Compile the `.tex` main file using PdfLaTeX with option `--enable-18`. This option is necessary to convert an expression in math mode to a static image in order to preserve its shape.[1]
    1.3. As result, we obtain a PDF file with a placeholder for each content insertion. The metadata is carried as hyperlink reference.
2. In this part we process the PDF file obtained in last step to obtain a HTML document in which each placeholder is replaced by its actual content.
    2.1. The PDF file is optimized using `Ghostscript` and converted to HTML using `pdf2htmlEX`.
    2.2. As result, we obtain an HTML and some auxiliary files. There's an `img` element for each placeholder and its metadata is carried in the `href` property.
    2.3. The HTML is processed by a Python application in which every content entry has its placeholder element replaced according to its class.[2]
    2.4. Finally we obtain our interactive document ready to be distributed over the web.

---

[1] If you are using a TeX editor, it may be possible to configure it to use this option automatically.

[2] Currently, only the class names 'iframe' and 'video' are supported. They store its `src` property as metadata and are replaced by an `iframe` element. Soon it will be possible for users to define their own replacement directives.

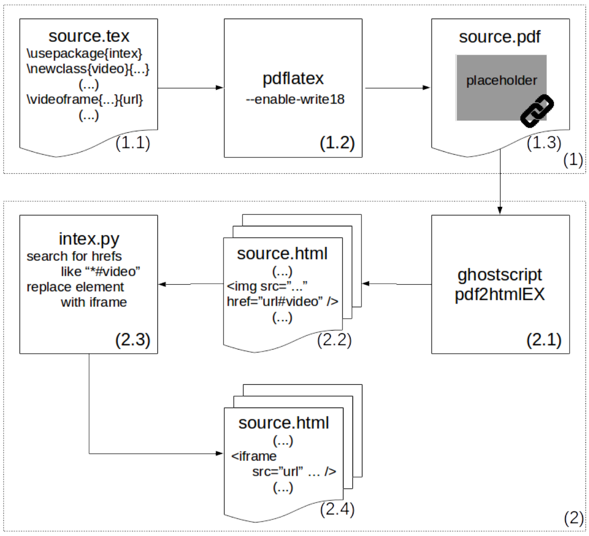


Figure 1: Workflow to generate an interactive document using InTeX

3 A working example

3.1 Setting up

Let's initiate a new class called `applet`.

Code 1: Initiating a new environment

```
\newclass{applet}{Applet}{List of applets}
```

Now we are able to include applet frames along the document. There are two approaches for doing this.

3.2 Inserting content

3.2.1 The easy way

If you are not a TeXskilled user nor familiar to InTeX, it's probably the best way for you to get started. There are a few short commands to insert content frames with default layout that should fit in most use cases. To insert an applet frame with empty placeholder and caption, we call:

Code 2: Frame with empty placeholder and caption

```
\appletframecap{.8}{.5}{https://...}{Theorem of Thales}
```

`https://www.geogebra.org/material/iframe/id/HjsJy8FV`

Applet 3.1: Theorem of Thales

The same frame can be obtained without frame with the command:

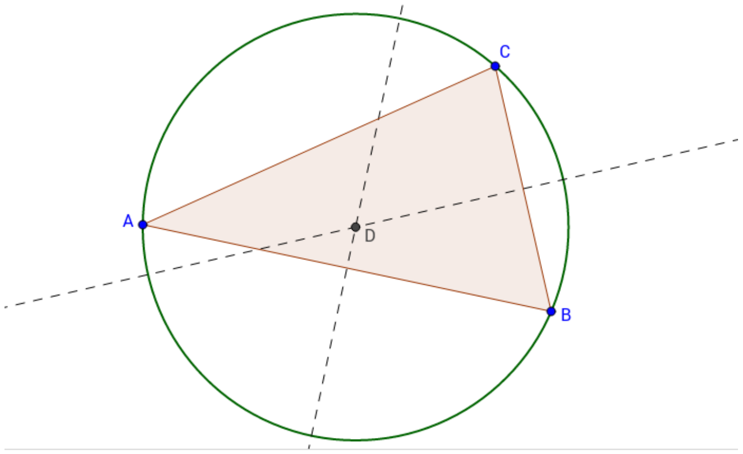
Code 3: Frame with empty placeholder and no caption

```
\appletframe{.8}{.5}{https://...}
```

Now let's say you want your document to be meaningful in its static version as well. Then you should consider using a `gframe` which employs pictures instead of empty placeholders. Although the images will remain static in the PDF version, they can point to a working url through a hyperlink reference. This kind of frames can be inserted with:

Code 4: Frame with image placeholder and caption

```
\appletgframecap[width=.8\textwidth]{ggb.png}{https://...}
{Circumcircle of a triangle}
```



Applet 3.2: Circumcircle of a triangle

Like the command for frames with empty placeholder, this one has also a variant without caption:

Code 5: Frame with image placeholder and no caption

```
\appletgframe[width=.8\textwidth]{ggb.png}{https://...}
```

3.2.2 The (not so) hard way

If the content frame is not placed where you want or you are not satisfied with its layout and appearence, you may prefer this way instead. Basically, it looks like including an image using `graphicx` inside a `figure` environment. The following lines are the expanded form of the commands presented before:

Code 6: Expanded form of Code 2

```
\begin{applet}
  \centering
  \includeapplet{.8\textwidth}{.5\textwidth}{https://...}
  \caption{Theorem of Thales}
\end{applet}
```

Code 7: Expanded form of Code 4

```
\begin{applet}
  \centering
  \includegapplet[width=.8\textwidth]{ggb.png}{https://...}
  \caption{Circumcircle of a triangle}
\end{applet}
```

Since `applet` is a `float` environment, it accepts the same options and behaves as any other environment of this type. The command `\includegapplet` accepts the same options as `\includegraphics` from `graphicx` package.

3.3 Making the list-of

You can easily make the table of contents for the `applet` environment by calling:

Code 8: List of applets

```
\listofapplet
```

List of applets

4 API reference

`\newclass{<name>}{<pretty name>}{<listof title>}` initializes a new class envirionment. `<name>` is used to name the environment and as prefix or sufix to compose its command names. `<pretty name>` is used in captions. `<listof title>` is used as title for the list index.

`\include<name>{<width>}{<height>}{<metadata>}` includes an empty placeholder with given dimentions and metadata.

`\includeg<name>[<opt>]{<img>}{<metadata>}` includes an image placeholder using `<img>` with given metadata. `<opt>` is passed as option to `\includegraphics`.

`\<name>frame{<swidth>}{<sheight>}{<metadata>}` calls `\include<name>` inside a centered frame with given arguments. The values `<swidth>` and `<sheight>` are scaled to `\textwidth`.

`\<name>framecap{<swidth>}{<sheight>}{<metadata>}{<caption>}` the same as `\<name>frame`, but with caption.

`\<name>gframe[<opt>]{<img>}{<metadata>}` calls `\includeg<name>` inside a centered frame with given arguments.

`\<name>gframe[<opt>]{<img>}{<metadata>}{<caption>}` the same as `\<name>gframe`, but with caption.

`\listof<name>` make the list of environment using default style.