

TeX in 2001

are we still up-to-date?

Competing with DTP

In desk top applications, text and graphic can be put anywhere on the page.

Fonts, spacing, color and all relevant aspects of text can be influenced.

T_EX is more restrictive in some aspects, or at least suggest less flexibility because macros try to be clever as well as provide limited functionality.

T_EX is a potential winner when it comes to combining graphics but designers don't think that way.

Desk top application come with good manuals, examples, courses and support.

001

to-date?

Competing with DTP

In desk top applications, text and graphic can be put anywhere on the page.

Fonts, spacing, color and all relevant aspects of text can be influenced.

T_EX is more restrictive in some aspects, or at least suggest less flexibility because macros try to be clever as well as provide limited functionality.

T_EX is a potential winner when it comes to combining graphics but designers don't think that way.

Desk top application come with good manuals, examples, courses and support.

001

to-date?

Competing with DTP

In desk top applications, text and graphic can be put anywhere on the page.

Fonts, spacing, color and all relevant aspects of text can be influenced.

T_EX is more restrictive in some aspects, or at least suggest less flexibility because macros try to be clever as well as provide limited functionality.

T_EX is a potential winner when it comes to combining graphics but designers don't think that way.

Desk top application come with good manuals, examples, courses and support.

001

to-date?

Competing with DTP

In desk top applications, text and graphic can be put anywhere on the page.

Fonts, spacing, color and all relevant aspects of text can be influenced.

T_EX is more restrictive in some aspects, or at least suggest less flexibility because macros try to be clever as well as provide limited functionality.

T_EX is a potential winner when it comes to combining graphics but designers don't think that way.

Desk top application come with good manuals, examples, courses and support.

001

to-date?

Competing with DTP

In desk top applications, text and graphic can be put anywhere on the page.

Fonts, spacing, color and all relevant aspects of text can be influenced.

T_EX is more restrictive in some aspects, or at least suggest less flexibility because macros try to be clever as well as provide limited functionality.

T_EX is a potential winner when it comes to combining graphics but designers don't think that way.

Desk top application come with good manuals, examples, courses and support.

001

to-date?

Competing with DTP

In desk top applications, text and graphic can be put anywhere on the page.

Fonts, spacing, color and all relevant aspects of text can be influenced.

T_EX is more restrictive in some aspects, or at least suggest less flexibility because macros try to be clever as well as provide limited functionality.

T_EX is a potential winner when it comes to combining graphics but designers don't think that way.

Desk top application come with good manuals, examples, courses and support.

001

to-date?

Competing with DTP

In desk top applications, text and graphics can be put anywhere on the page.

Fonts, spacing, color and all relevant text can be influenced.

TEX is more restrictive in some aspects suggest less flexibility because macros are clever as well as provide limited functions.

TEX is a potential winner when it comes to combining graphics but designers don't know the way.

Desk top applications come with good examples, courses and support.

Conceptual Limitations

If you want batch, you need to catch a lot of border cases in advance.

If you open a possibility, you also provide an opening to inconsistency and (in case of a reprogrammable system) misuse.

But, since some control is wanted, the impossible is often not needed. So, providing structured control also provides stability.

TEX-like tools at first look slow, demanding, old fashioned but when introduced and supported properly can gain lifelong sympathy.

Competing with DTP

In desk top applications, text and graphics are put anywhere on the page.

Fonts, spacing, color and all relevant text can be influenced.

TEX is more restrictive in some aspects suggest less flexibility because macros are clever as well as provide limited functions.

TEX is a potential winner when it comes to combining graphics but designers don't know the way.

Desk top applications come with good examples, courses and support.

Conceptual Limitations

If you want batch, you need to catch a lot of border cases in advance.

If you open a possibility, you also provide an opening to inconsistency and (in case of a reprogrammable system) misuse.

But, since some control is wanted, the impossible is often not needed. So, providing structured control also provides stability.

TEX-like tools at first look slow, demanding, old fashioned but when introduced and supported properly can gain lifelong sympathy.

Competing with DTP

In desk top applications, text and graphics are put anywhere on the page.

Fonts, spacing, color and all relevant text can be influenced.

TEX is more restrictive in some aspects suggest less flexibility because macros are clever as well as provide limited functions.

TEX is a potential winner when it comes to combining graphics but designers don't know the way.

Desk top applications come with good examples, courses and support.

Conceptual Limitations

If you want batch, you need to catch a lot of border cases in advance.

If you open a possibility, you also provide an opening to inconsistency and (in case of a reprogrammable system) misuse.

But, since some control is wanted, the impossible is often not needed. So, providing structured control also provides stability.

TEX-like tools at first look slow, demanding, old fashioned but when introduced and supported properly can gain lifelong sympathy.

Competing with DTP

In desk top applications, text and graphics are put anywhere on the page.

Fonts, spacing, color and all relevant text can be influenced.

TEX is more restrictive in some aspects suggest less flexibility because macros are clever as well as provide limited functions.

TEX is a potential winner when it comes to combining graphics but designers don't know the way.

Desk top applications come with good examples, courses and support.

Conceptual Limitations

If you want batch, you need to catch a lot of border cases in advance.

If you open a possibility, you also provide an opening to inconsistency and (in case of a reprogrammable system) misuse.

But, since some control is wanted, the impossible is often not needed. So, providing structured control also provides stability.

TEX-like tools at first look slow, demanding, old fashioned but when introduced and supported properly can gain lifelong sympathy.

Competing with DTP

In desk top applications, text and graphics can be put anywhere on the page.

Fonts, spacing, color and all relevant text can be influenced.

T_EX is more restrictive in some aspects suggest less flexibility because macros are clever as well as provide limited functions.

T_EX is a potential winner when it comes to combining graphics but designers don't know the way.

Desk top applications come with good examples, courses and support.

Conceptual Limitations

If you want batch, you need to catch a lot of border cases in advance.

If you open a possibility, you also provide an opening to inconsistency and (in case of a reprogrammable system) misuse.

But, since some control is wanted, the impossible is often not needed. So, providing structured control also provides stability.

T_EX-like tools at first look slow, demanding, old fashioned but when introduced and supported properly can gain lifelong sympathy.

Competing with DTP

In desk top applications, text and graphics can be placed anywhere on the page.

Fonts, spacing, color and all relevant text can be influenced.

TeX is more restrictive in some aspects suggest less flexibility because macros are clever as well as provide limited functions.

TeX is a potential winner when it comes to combining graphics but designers don't know the way.

Desk top applications come with good examples, courses and support.

Conceptual Limitations

If you want batch, you need to catch a lot of border cases in advance.

If you open a possibility, you also provide an opening to inconsistency and (in case of a reprogrammable system) misuse.

But, since some control is wanted, the impossible is often not needed. So, providing structured control also provides stability.

TeX-like tools at first look slow, demanding, old fashioned but when introduced and supported properly can gain lifelong sympathy.

Competing with Authoring

Documents become more than alone text, so we need to provide interactive elements.

Depending on the technology used, such elements are fragile and not safe for the future.

So, at the same time we need to guard the traditional means of navigating documents.

Here, TeX can often go further than any other system, simply because it's programmable.

Competing with DTP

In desk top applications, text and graphics can be placed anywhere on the page.

Fonts, spacing, color and all relevant text can be influenced.

TeX is more restrictive in some aspects suggest less flexibility because macros are clever as well as provide limited functions.

TeX is a potential winner when it comes to combining graphics but designers don't know the way.

Desk top applications come with good examples, courses and support.

Conceptual Limitations

If you want batch, you need to catch a lot of border cases in advance.

If you open a possibility, you also provide an opening to inconsistency and (in case of a reprogrammable system) misuse.

But, since some control is wanted, the impossible is often not needed. So, providing structured control also provides stability.

TeX-like tools at first look slow, demanding, old fashioned but when introduced and supported properly can gain lifelong sympathy.

Competing with Authoring

Documents become more than just text, so we need to provide interactive elements.

Depending on the technology used, such elements are fragile and not safe for the future.

So, at the same time we need to guard the traditional means of navigating documents.

Here, TeX can often go further than any other system, simply because it's programmable.

Competing with DTP

In desk top applications, text and graphics can be placed anywhere on the page.

Fonts, spacing, color and all relevant text can be influenced.

TeX is more restrictive in some aspects suggest less flexibility because macros are clever as well as provide limited functions.

TeX is a potential winner when it comes to combining graphics but designers don't know the way.

Desk top applications come with good examples, courses and support.

Conceptual Limitations

If you want batch, you need to catch a lot of border cases in advance.

If you open a possibility, you also provide an opening to inconsistency and (in case of a reprogrammable system) misuse.

But, since some control is wanted, the impossible is often not needed. So, providing structured control also provides stability.

TeX-like tools at first look slow, demanding, old fashioned but when introduced and supported properly can gain lifelong sympathy.

Competing with Authoring

Documents become more than just text, so we need to provide interactive elements.

Depending on the technology used, such elements are fragile and not safe for the future.

So, at the same time we need to guard the traditional means of navigating documents.

Here, TeX can often go further than any other system, simply because it's programmable.

Competing with DTP

In desk top applications, text and graphics are input anywhere on the page.

Fonts, spacing, color and all relevant text can be influenced.

T_EX is more restrictive in some aspects suggest less flexibility because macros are clever as well as provide limited functions.

T_EX is a potential winner when it comes to combining graphics but designers don't know the way.

Desk top applications come with good examples, courses and support.

Conceptual Limitations

If you want batch, you need to catch a lot of border cases in advance.

If you open a possibility, you also provide an opening to inconsistency and (in case of a reprogrammable system) misuse.

But, since some control is wanted, the impossible is often not needed. So, providing structured control also provides stability.

T_EX-like tools at first look slow, demanding, old fashioned but when introduced and supported properly can gain lifelong sympathy.

Competing with Authoring

Documents become more than alone text, so we need to provide interactive elements.

Depending on the technology used, such elements are fragile and not safe for the future.

So, at the same time we need to guard the traditional means of navigating documents.

Here, T_EX can often go further than any other system, simply because it's programmable.

Competing with DTP

In desk top applications, text and graphics can be placed anywhere on the page.

Fonts, spacing, color and all relevant text can be influenced.

T_EX is more restrictive in some aspects suggest less flexibility because macros are clever as well as provide limited functions.

T_EX is a potential winner when it comes to combining graphics but designers don't know the way.

Desk top applications come with good examples, courses and support.

Conceptual Limitations

If you want batch, you need to catch a lot of border cases in advance.

If you open a possibility, you also provide an opening to inconsistency and (in case of a reprogrammable system) misuse.

But, since some control is wanted, the impossible is often not needed. So, providing structured control also provides stability.

T_EX-like tools at first look slow, demanding, old fashioned but when introduced and supported properly can gain lifelong sympathy.

Competing with Authoring

Documents become more than alone text, so we need to provide interactive elements.

Depending on the technology used, such elements are fragile and not safe for the future.

So, at the same time we need to guard the traditional means of navigating documents.

Here, T_EX can often go further than any other system, simply because it's programmable.

Competing with DTP

In desk top applications, text and graphics are put anywhere on the page.

Fonts, spacing, color and all relevant text can be influenced.

\TeX is more restrictive in some aspects suggest less flexibility because macros are clever as well as provide limited functions.

\TeX is a potential winner when it comes

Competing with Fashion

There is a difference between making a rough sketch and a real product.

A proof of concept is not per definition a proof of usability.

When properly supported, and accompanied with the right tools and methods, demanding authors are willing to use \TeX .

We have to make sure that \TeX can do most of the (decent) things that designers want to do.

We also need to educate designers in automatic document processing: its weaknesses and strengths.

Conceptual Limitations

If you want batch, you need to catch a lot of border cases in advance.

If you open a possibility, you also provide an opening to inconsistency and (in case of a reprogrammable system) misuse.

But, since some control is wanted, the impossible is often not needed. So, providing structured control also provides stability.

It first look slow, demanding, old-fashioned when introduced and supported popular sympathy.

Authoring

More than alone text, so we need to support graphics.

Technology used, such elements are fragile and error-prone.

We need to guard the traditional means.

Other than any other system, simply better.

Competing with DTP

In desk top applications, text and graphics are put anywhere on the page.

Fonts, spacing, color and all relevant text can be influenced.

\TeX is more restrictive in some aspects suggest less flexibility because macros are clever as well as provide limited functions.

\TeX is a potential winner when it comes to

Competing with Fashion

There is a difference between making a rough sketch and a real product.

A proof of concept is not per definition a proof of usability.

When properly supported, and accompanied with the right tools and methods, demanding authors are willing to use \TeX .

We have to make sure that \TeX can do most of the (decent) things that designers want to do.

We also need to educate designers in automatic document processing: its weaknesses and strengths.

Conceptual Limitations

If you want batch, you need to catch a lot of border cases in advance.

If you open a possibility, you also provide an opening to inconsistency and (in case of a reprogrammable system) misuse.

But, since some control is wanted, the impossible is often not needed. So, providing structured control also provides stability.

It first look slow, demanding, old-fashioned and not well introduced and supported program, but it has a long sympathy.

Authoring

Authoring is more than alone text, so we need to support graphics.

Technology used, such elements are fragile and need to be protected.

We need to guard the traditional means of authoring.

Compared to other systems, \TeX is rather than any other system, simply better.

Competing with DTP

In desk top applications, text and graphics are put anywhere on the page.

Fonts, spacing, color and all relevant text can be influenced.

\TeX is more restrictive in some aspects suggest less flexibility because macros are clever as well as provide limited functions.

\TeX is a potential winner when it comes to

Competing with Fashion

There is a difference between making a rough sketch and a real product.

A proof of concept is not per definition a proof of usability.

When properly supported, and accompanied with the right tools and methods, demanding authors are willing to use \TeX .

We have to make sure that \TeX can do most of the (decent) things that designers want to do.

We also need to educate designers in automatic document processing: its weaknesses and strengths.

Conceptual Limitations

If you want batch, you need to catch a lot of border cases in advance.

If you open a possibility, you also provide an opening to inconsistency and (in case of a reprogrammable system) misuse.

But, since some control is wanted, the impossible is often not needed. So, providing structured control also provides stability.

It first look slow, demanding, old-fashioned when introduced and supported popular long sympathy.

Authoring

More than alone text, so we need to support graphics.

Technology used, such elements are fragile and expensive.

We need to guard the traditional means.

Other than any other system, simply better.

Competing with DTP

In desk top applications, text and graphics are put anywhere on the page.

Fonts, spacing, color and all relevant text can be influenced.

\TeX is more restrictive in some aspects, suggest less flexibility because macros are clever as well as provide limited functions.

\TeX is a potential winner when it comes to

Competing with Fashion

There is a difference between making a rough sketch and a real product.

A proof of concept is not per definition a proof of usability.

When properly supported, and accompanied with the right tools and methods, demanding authors are willing to use \TeX .

We have to make sure that \TeX can do most of the (decent) things that designers want to do.

We also need to educate designers in automatic document processing: its weaknesses and strengths.

Conceptual Limitations

If you want batch, you need to catch a lot of border cases in advance.

If you open a possibility, you also provide an opening to inconsistency and (in case of a reprogrammable system) misuse.

But, since some control is wanted, the impossible is often not needed. So, providing structured control also provides stability.

It first look slow, demanding, old-fashioned when introduced and supported popular long sympathy.

Authoring

More than alone text, so we need to support graphics.

Technology used, such elements are fragile and expensive.

We need to guard the traditional means.

Other than any other system, simply better.

Competing with DTP

In desk top applications, text and graphics are put anywhere on the page.

Fonts, spacing, color and all relevant text can be influenced.

\TeX is more restrictive in some aspects suggest less flexibility because macros are clever as well as provide limited functions.

\TeX is a potential winner when it comes to

Competing with Fashion

There is a difference between making a rough sketch and a real product.

A proof of concept is not per definition a proof of usability.

When properly supported, and accompanied with the right tools and methods, demanding authors are willing to use \TeX .

We have to make sure that \TeX can do most of the (decent) things that designers want to do.

We also need to educate designers in automatic document processing: its weaknesses and strengths.

Conceptual Limitations

If you want batch, you need to catch a lot of border cases in advance.

If you open a possibility, you also provide an opening to inconsistency and (in case of a reprogrammable system) misuse.

But, since some control is wanted, the impossible is often not needed. So, providing structured control also provides stability.

It first look slow, demanding, old-fashioned when introduced and supported popular long sympathy.

Authoring

More than alone text, so we need to support graphics.

Technology used, such elements are fragile and error-prone.

We need to guard the traditional means.

Other than any other system, simply better.

Competing with DTP

In desk top applications, text and graphics are put anywhere on the page.

Fonts, spacing, color and all relevant text can be influenced.

\TeX is more restrictive in some aspects suggest less flexibility because macros are clever as well as provide limited functions.

\TeX is a potential winner when it comes to

Competing with Fashion

There is a difference between making a rough sketch and a real product.

A proof of concept is not per definition a proof of usability.

When properly supported, and accompanied with the right tools and methods, demanding authors are willing to use \TeX .

We have to make sure that \TeX can do most of the (decent) things that designers want to do.

We also need to educate designers in automatic document processing: its weaknesses and strengths.

Conceptual Limitations

If you want batch, you need to catch a lot of border cases in advance.

If you open a possibility, you also provide an opening to inconsistency and (in case of a reprogrammable system) misuse.

But, since some control is wanted, the impossible is often not needed. So, providing structured control also provides stability.

It first look slow, demanding, old-fashioned when introduced and supported professional sympathy.

Authoring

More than alone text, so we need to support graphics.

Technology used, such elements are fragile and error-prone.

We need to guard the traditional means.

Other than any other system, simply not.

Publisher Demands

For long, \TeX was the only affordable choice for typesetting math, so "anything was better than nothing". But times have changed.

More and more, graphics and color becomes natural in print an on desktops. What you see elsewhere, is what you want yourself.

We need to provide DTP competing output with a batch oriented program. Here we can use \TeX 's strength to provide multiple products.

Since "everyone can be a typesetter", the budgets for design and production of documents are relatively low (even upto the unacceptable). But, reuse pays off.

Step by isolated step production is replaced by integrated workflows. The often tight schedules demand robust methods.

Since publishers want to be in business, we must keep an eye on XML and \TeX can handle that quite well. We can even benefit from it.

(recent) things that designers want to do.

We also need to educate designers in automatic document processing: its weaknesses and strength.

Qual Limitations

*batch, you need to catch a lot of border
ance.*

*a possibility, you also provide an opening
ency and (in case of a reprogrammable
use.*

*ome control is wanted, the impossible is
eded. So, providing structured control
s stability.*

*t first look slow, demanding, old
ien introduced and supported prop-
long sympathy.*

Authoring

*than alone text, so we need to
ts.*

*he
ng
e. technology used, such elements are fragile
e.*

he need to guard the traditional means

rather than any other system, simply

Publisher Demands

For long, \TeX was the only affordable choice for typesetting math, so “anything was better than nothing”. But times have changed.

More and more, graphics and color becomes natural in print an on desktops. What you see elsewhere, is what you want yourself.

We need to provide DTP competing output with a batch oriented program. Here we can use \TeX 's strength to provide multiple products.

Since “everyone can be a typesetter”, the budgets for design and production of documents are relatively low (even upto the unacceptable). But, reuse pays off.

Step by isolated step production is replaced by integrated workflows. The often tight schedules demand robust methods.

Since publishers want to be in business, we must keep an eye on XML and \TeX can handle that quite well. We can even benefit from it.

(recent) things that designers want to do.

We also need to educate designers in automatic document processing: its weaknesses and strength.

Qual Limitations

*batch, you need to catch a lot of border
ance.*

*a possibility, you also provide an opening
ency and (in case of a reprogrammable
use.*

*ome control is wanted, the impossible is
eded. So, providing structured control
s stability.*

*t first look slow, demanding, old
ien introduced and supported prop-
long sympathy.*

Authoring

*of
than alone text, so we need to
ts.*

*he
ng
ogy used, such elements are fragile
e.*

*he
eed to guard the traditional means*

rather than any other system, simply

Publisher Demands

For long, \TeX was the only affordable choice for typesetting math, so “anything was better than nothing”. But times have changed.

More and more, graphics and color becomes natural in print and on desktops. What you see elsewhere, is what you want yourself.

We need to provide DTP competing output with a batch oriented program. Here we can use \TeX 's strength to provide multiple products.

Since “everyone can be a typesetter”, the budgets for design and production of documents are relatively low (even upto the unacceptable). But, reuse pays off.

Step by isolated step production is replaced by integrated workflows. The often tight schedules demand robust methods.

Since publishers want to be in business, we must keep an eye on XML and \TeX can handle that quite well. We can even benefit from it.

(Recent) things that designers want to do.

We also need to educate designers in automatic document processing: its weaknesses and strength.

Qual Limitations

*batch, you need to catch a lot of border
ance.*

*possibility, you also provide an opening
ency and (in case of a reprogrammable
use.*

*ome control is wanted, the impossible is
eded. So, providing structured control
s stability.*

*t first look slow, demanding, old
ien introduced and supported prop-
long sympathy.*

Authoring

*than alone text, so we need to
ts.*

*he
ng
e. technology used, such elements are fragile
e.*

he need to guard the traditional means

rather than any other system, simply

Publisher Demands

For long, \TeX was the only affordable choice for typesetting math, so “anything was better than nothing”. But times have changed.

More and more, graphics and color becomes natural in print an on desktops. What you see elsewhere, is what you want yourself.

We need to provide DTP competing output with a batch oriented program. Here we can use \TeX 's strength to provide multiple products.

Since “everyone can be a typesetter”, the budgets for design and production of documents are relatively low (even upto the unacceptable). But, reuse pays off.

Step by isolated step production is replaced by integrated workflows. The often tight schedules demand robust methods.

Since publishers want to be in business, we must keep an eye on XML and \TeX can handle that quite well. We can even benefit from it.

(recent) things that designers want to do.

We also need to educate designers in automatic document processing: its weaknesses and strength.

Qual Limitations

*batch, you need to catch a lot of border
ance.*

*possibility, you also provide an opening
ency and (in case of a reprogrammable
use.*

*ome control is wanted, the impossible is
eded. So, providing structured control
s stability.*

*t first look slow, demanding, old
ien introduced and supported prop-
long sympathy.*

Authoring

*than alone text, so we need to
ts.*

*he
ng
e. technology used, such elements are fragile*

he need to guard the traditional means

rather than any other system, simply

Publisher Demands

For long, \TeX was the only affordable choice for typesetting math, so “anything was better than nothing”. But times have changed.

More and more, graphics and color becomes natural in print an on desktops. What you see elsewhere, is what you want yourself.

We need to provide DTP competing output with a batch oriented program. Here we can use \TeX 's strength to provide multiple products.

Since “everyone can be a typesetter”, the budgets for design and production of documents are relatively low (even upto the unacceptable). But, reuse pays off.

Step by isolated step production is replaced by integrated workflows. The often tight schedules demand robust methods.

Since publishers want to be in business, we must keep an eye on XML and \TeX can handle that quite well. We can even benefit from it.

(recent) things that designers want to do.

We also need to educate designers in automatic document processing: its weaknesses and strength.

Qual Limitations

*batch, you need to catch a lot of border
ance.*

*possibility, you also provide an opening
ency and (in case of a reprogrammable
use.*

*ome control is wanted, the impossible is
eded. So, providing structured control
s stability.*

*t first look slow, demanding, old
ien introduced and supported prop-
long sympathy.*

Authoring

*of than alone text, so we need to
ts.*

*he technology used, such elements are fragile
e.*

he need to guard the traditional means

rather than any other system, simply

Publisher Demands

For long, \TeX was the only affordable choice for typesetting math, so “anything was better than nothing”. But times have changed.

More and more, graphics and color becomes natural in print an on desktops. What you see elsewhere, is what you want yourself.

We need to provide DTP competing output with a batch oriented program. Here we can use \TeX 's strength to provide multiple products.

Since “everyone can be a typesetter”, the budgets for design and production of documents are relatively low (even upto the unacceptable). But, reuse pays off.

Step by isolated step production is replaced by integrated workflows. The often tight schedules demand robust methods.

Since publishers want to be in business, we must keep an eye on XML and \TeX can handle that quite well. We can even benefit from it.

(Recent) things that designers want to do.

We also need to educate designers in automatic document processing: its weaknesses and strength.

Qual Limitations

*batch, you need to catch a lot of border
ance.*

*possibility, you also provide an opening
ency and (in case of a reprogrammable
use.*

*ome control is wanted, the impossible is
eded. So, providing structured control
s stability.*

*t first look slow, demanding, old
ien introduced and supported prop-
long sympathy.*

Authoring

*of than alone text, so we need to
ts.*

*he ts.
ng ology used, such elements are fragile
e.*

he eed to guard the traditional means

rather than any other system, simply

Publisher Demands

For long, \TeX was the only affordable choice for typesetting math, so “anything was better than nothing”. But times have changed.

More and more, graphics and color becomes natural in print and on desktops. What you see elsewhere, is what you want yourself.

We need to provide DTP competing output with a batch oriented program. Here we can use \TeX 's strength to provide multiple products.

Since “everyone can be a typesetter”, the budgets for design and production of documents are relatively low (even upto the unacceptable). But, reuse pays off.

Step by isolated step production is replaced by integrated workflows. The often tight schedules demand robust methods.

Since publishers want to be in business, we must keep an eye on XML and \TeX can handle that quite well. We can even benefit from it.

We also need to educate designers in automatic document processing: its weaknesses and strength.

Qual Limitations

*batch, you need to catch a lot of border
ance.*

*possibility, you also provide an opening
ency and (in case of a reprogrammable
use.*

*ome control is wanted, the impossible is
eded. So, providing structured control
s stability.*

*t first look slow, demanding, old
ien introduced and supported prop-
long sympathy.*

Authoring

*of
than alone text, so we need to
ts.*

*he
ng
ogy used, such elements are fragile
e.*

*he
eed to guard the traditional means*

rather than any other system, simply

Publisher Demands

For long, \TeX was the only typesetting math, so “anything but \TeX is nothing”. But times have changed.

More and more, graphics and typesetting are done in print and on desktops. What is what you want yourself.

We need to provide DTP capabilities, a batch oriented program. It has the strength to provide multiple users.

Since “everyone can be a typesetter” for design and production, the price is actively low (even upto the point where reuse pays off).

Step by isolated step production is integrated workflows. The demand robust methods.

Since publishers want to be in business, we must keep an eye on XML and \TeX can handle that quite well. We can even benefit from it.

We also need to educate designers in automatic document processing: its weaknesses and strengths.

Author Demands

Authors take part in the preparation of the final products. Not seldom, they produce most of the product.

In order to have some control, publishers can provide the tools, styles and/or strict rules.

If a separate design department is used, even not so complicated (math) texts can result in endless feedback-loops.

Being a wide spread tool, is no guarantee for being a good tool (this is true for \TeX as well as commercial applications). Tools get kicked out on and off.

Authors still want a certain level of control, but if the results are great, they are willing to accept limitations.

Authors often have a clear picture of what they want to see in print, and today they print on their desktop.

In serving the authors, we must not forget the demands of publishers, especially the need for reuse of source code.

...e.
...he need to guard the traditional means

...rther than any other system, simply

Publisher Demands

For long, \TeX was the only typesetting math, so “anything but \TeX is nothing”. But times have changed.

More and more, graphics and typesetting are done in print and on desktops. What you want is what you want yourself.

We need to provide DTP capabilities, not just a batch oriented program. We need the strength to provide multiple users.

Since “everyone can be a typographer”, the demand for design and production tools is naturally low (even upto the point where reuse pays off).

Step by isolated step production is not an integrated workflow. The demand is for robust methods.

Since publishers want to be in business, we must keep an eye on XML and \TeX can handle that quite well. We can even benefit from it.

We also need to educate designers in automatic document processing: its weaknesses and strengths.

Author Demands

Authors take part in the preparation of the final products. Not seldom, they produce most of the product.

In order to have some control, publishers can provide the tools, styles and/or strict rules.

If a separate design department is used, even not so complicated (math) texts can result in endless feedback-loops.

Being a wide spread tool, is no guarantee for being a good tool (this is true for \TeX as well as commercial applications). Tools get kicked out on and off.

Authors still want a certain level of control, but if the results are great, they are willing to accept limitations.

Authors often have a clear picture of what they want to see in print, and today they print on their desktop.

In serving the authors, we must not forget the demands of publishers, especially the need for reuse of source code.

...e.
...he need to guard the traditional means

...rther than any other system, simply

Publisher Demands

For long, \TeX was the only typesetting math, so “anything goes”. But times have changed.

More and more, graphics and layout are done in print and on desktops. What you want is what you want yourself.

We need to provide DTP capabilities, not a batch oriented program. We need the strength to provide multiple users.

Since “everyone can be a typographer”, the demand for design and production is becoming increasingly low (even upto the point where reuse pays off).

Step by isolated step production is being replaced by integrated workflows. The demand for robust methods.

Since publishers want to be in business, we must keep an eye on XML and \TeX can handle that quite well. We can even benefit from it.

We also need to educate designers in automatic document processing: its weaknesses and strengths.

Author Demands

Authors take part in the preparation of the final products. Not seldom, they produce most of the product.

In order to have some control, publishers can provide the tools, styles and/or strict rules.

If a separate design department is used, even not so complicated (math) texts can result in endless feedback-loops.

Being a wide spread tool, is no guarantee for being a good tool (this is true for \TeX as well as commercial applications). Tools get kicked out on and off.

Authors still want a certain level of control, but if the results are great, they are willing to accept limitations.

Authors often have a clear picture of what they want to see in print, and today they print on their desktop.

In serving the authors, we must not forget the demands of publishers, especially the need for reuse of source code.

...e.
...he need to guard the traditional means

...rther than any other system, simply

Publisher Demands

For long, \TeX was the only typesetting math, so “anything goes”. But times have changed.

More and more, graphics and layout are done in print and on desktops. What you want is what you want yourself.

We need to provide DTP capabilities, a batch oriented program, and the strength to provide multiple users.

Since “everyone can be a typographer” for design and production, the price is actively low (even upto the point where reuse pays off).

Step by isolated step production is not integrated workflows. The demand is for robust methods.

Since publishers want to be in business, we must keep an eye on XML and \TeX can handle that quite well. We can even benefit from it.

We also need to educate designers in automatic document processing: its weaknesses and strengths.

Author Demands

Authors take part in the preparation of the final products. Not seldom, they produce most of the product.

In order to have some control, publishers can provide the tools, styles and/or strict rules.

If a separate design department is used, even not so complicated (math) texts can result in endless feedback-loops.

Being a wide spread tool, is no guarantee for being a good tool (this is true for \TeX as well as commercial applications). Tools get kicked out on and off.

Authors still want a certain level of control, but if the results are great, they are willing to accept limitations.

Authors often have a clear picture of what they want to see in print, and today they print on their desktop.

In serving the authors, we must not forget the demands of publishers, especially the need for reuse of source code.

...e.
...he need to guard the traditional means

...rther than any other system, simply

Publisher Demands

For long, \TeX was the only typesetting math, so “anything goes”. But times have changed.

More and more, graphics and layout are done in print and on desktops. What you want is what you want yourself.

We need to provide DTP capabilities, a batch oriented program, and the strength to provide multiple outputs.

Since “everyone can be a typographer” for design and production, the price is actively low (even upto the point where reuse pays off).

Step by isolated step production is not integrated workflows. The demand is for robust methods.

Since publishers want to be in business, we must keep an eye on XML and \TeX can handle that quite well. We can even benefit from it.

We also need to educate designers in automatic document processing: its weaknesses and strengths.

Author Demands

Authors take part in the preparation of the final products. Not seldom, they produce most of the product.

In order to have some control, publishers can provide the tools, styles and/or strict rules.

If a separate design department is used, even not so complicated (math) texts can result in endless feedback-loops.

Being a wide spread tool, is no guarantee for being a good tool (this is true for \TeX as well as commercial applications). Tools get kicked out on and off.

Authors still want a certain level of control, but if the results are great, they are willing to accept limitations.

Authors often have a clear picture of what they want to see in print, and today they print on their desktop.

In serving the authors, we must not forget the demands of publishers, especially the need for reuse of source code.

...e.
...he need to guard the traditional means

...rther than any other system, simply

Publisher Demands

For long, \TeX was the only typesetting math, so “anything but \TeX is nothing”. But times have changed.

More and more, graphics and typesetting are done in print and on desktops. What is what you want yourself.

We need to provide DTP capabilities in a batch oriented program. It is a strength to provide multiple outputs.

Since “everyone can be a typesetter”, the demand for design and production is becoming increasingly low (even upto the point where reuse pays off).

Step by isolated step production is being replaced by integrated workflows. The demand for robust methods.

Since publishers want to be in business, we must keep an eye on XML and \TeX can handle that quite well. We can even benefit from it.

(recent) things that designers want to do.

We also need to educate designers in automatic document processing: its weaknesses and strengths.

Author Demands

Authors take part in the preparation of the final products. Not seldom, they produce most of the product.

In order to have some control, publishers can provide the tools, styles and/or strict rules.

If a separate design department is used, even not so complicated (math) texts can result in endless feedback-loops.

Being a wide spread tool, is no guarantee for being a good tool (this is true for \TeX as well as commercial applications). Tools get kicked out on and off.

Authors still want a certain level of control, but if the results are great, they are willing to accept limitations.

Authors often have a clear picture of what they want to see in print, and today they print on their desktop.

In serving the authors, we must not forget the demands of publishers, especially the need for reuse of source code.

...e.
...he need to guard the traditional means

...rther than any other system, simply

Publisher Demands

For long, \TeX was the only typesetting math, so “anything but \TeX is nothing”. But times have changed.

More and more, graphics and layout are done in print and on desktops. What is what you want yourself.

We need to provide DTP capabilities as a batch oriented program. It must have the strength to provide multiple users.

Since “everyone can be a typographer” for design and production, the price is actively low (even upto the point where reuse pays off).

Step by isolated step production is not integrated workflows. The demand robust methods.

Since publishers want to be in business, we must keep an eye on XML and \TeX can handle that quite well. We can even benefit from it.

We also need to educate designers in automatic document processing: its weaknesses and strengths.

Author Demands

Authors take part in the preparation of the final products. Not seldom, they produce most of the product.

In order to have some control, publishers can provide the tools, styles and/or strict rules.

If a separate design department is used, even not so complicated (math) texts can result in endless feedback-loops.

Being a wide spread tool, is no guarantee for being a good tool (this is true for \TeX as well as commercial applications). Tools get kicked out on and off.

Authors still want a certain level of control, but if the results are great, they are willing to accept limitations.

Authors often have a clear picture of what they want to see in print, and today they print on their desktop.

In serving the authors, we must not forget the demands of publishers, especially the need for reuse of source code.

...e.
...he need to guard the traditional means

...rther than any other system, simply

Publisher Demands

For long, \TeX was the only typesetting math, so “anything but \TeX is nothing”. But times have changed.

More and more, graphics and typesetting are done in print and on desktops. What is what you want yourself.

We need to provide DTP capabilities in a batch oriented program. It is a strength to provide multiple users.

Since “everyone can be a typesetter”, the demand for design and production is becoming increasingly low (even upto the point where reuse pays off).

Step by isolated step production is being replaced by integrated workflows. The demand for robust methods.

Since publishers want to be in business, we must keep an eye on XML and \TeX can handle that quite well. We can even benefit from it.

(recent) things that designers want to do.

We also need to educate designers in automatic document processing: its weaknesses and strengths.

Author Demands

Authors take part in the preparation of the final products. Not seldom, they produce most of the product.

In order to have some control, publishers can provide the tools, styles and/or strict rules.

If a separate design department is used, even not so complicated (math) texts can result in endless feedback-loops.

Being a wide spread tool, is no guarantee for being a good tool (this is true for \TeX as well as commercial applications). Tools get kicked out on and off.

Authors still want a certain level of control, but if the results are great, they are willing to accept limitations.

Authors often have a clear picture of what they want to see in print, and today they print on their desktop.

In serving the authors, we must not forget the demands of publishers, especially the need for reuse of source code.

...e.
...he need to guard the traditional means

...rather than any other system, simply

Publisher Demands

For long, \TeX was the only typesetting math, so “anything goes”. But times have changed.

More and more, graphics and layout are done in print and on desktops. What you want is what you want yourself.

We need to provide DTP capabilities, not a batch oriented program. We need the strength to provide multiple outputs.

Since “everyone can be a typographer”, the demand for design and production tools is relatively low (even upto the point where reuse pays off).

Step by isolated step production is not integrated workflows. The demand is for robust methods.

Since publishers want to be in business, they will keep an eye on XML and \TeX can compete quite well. We can even benefit from the success of other (recent) things that designers want to use.

We also need to educate designers in the use of document processing: its weaknesses and strengths.

Author Demands

Authors take part in the preparation of the final products. Not seldom, they produce most of the product.

In order to have some control, publishers can provide the tools, styles and/or strict rules.

If a separate design department is used, even not so complicated (not a separate design department).

Being a tool (the user is the user). Tools get the user.

Authors are great.

Authors in print.

In service of publishers.

User Demands

\TeX is also used by non-profit organizations, home users, those who are on their own, and those who want control (like science students).

They want access to the whole bag of tricks, in a controlled way (and in \TeX even using fonts is trickery).

Since they are willing to participate in experiments, this provides an ideal playground for new developments.

A bag of latest tricks can endanger a stable system, and in this \TeX is not different from other applications.

Apart from trickery we can try to tap the power of file formats as much as possible.

Publisher Demands

For long, \TeX was the only typesetting math, so “anything but nothing”. But times have changed.

More and more, graphics and layout are done in print and on desktops. What is what you want yourself.

We need to provide DTP capabilities as a batch oriented program. It has the strength to provide multiple

Since “everyone can be a typesetter” for design and production, the cost is relatively low (even upto the reuse pays off).

Step by isolated step production is integrated workflows. The demand robust methods.

Since publishers want to be in business, keep an eye on XML and \TeX can do quite well. We can even benefit from

We also need to educate designers in document processing: its weaknesses and

Author Demands

Authors take part in the preparation of the final products. Not seldom, they produce most of the product.

In order to have some control, publishers can provide the tools, styles and/or strict rules.

If a separate design department is used, even not so complicated (not so much of a “trickery”)

Being a tool (the Tools get

Authors are great

Authors in print

In service publishers

User Demands

\TeX is also also used by non-profit organizations, home users, those who are on their own, and those who want control (like science students).

They want access to the whole bag of tricks, in a controlled way (and in \TeX even using fonts is trickery).

Since they are willing to participate in experiments, this provides an ideal playground for new developments.

A bag of latest tricks can endanger a stable system, and in this \TeX is not different from other applications.

Apart from trickery we can try to tap the power of file formats as much as possible.

Publisher Demands

For long, \TeX was the only typesetting math, so “anything goes”. But times have changed.

More and more, graphics and layout are done in print and on desktops. What you want is what you want yourself.

We need to provide DTP capabilities, not just a batch oriented program. We need the strength to provide multiple outputs.

Since “everyone can be a typographer” for design and production, the cost is relatively low (even upto the point where reuse pays off).

Step by isolated step production is not integrated workflows. The demand is for robust methods.

Since publishers want to be in business, they will keep an eye on XML and \TeX can do quite well. We can even benefit from it.

We also need to educate designers in document processing: its weaknesses and strengths.

Author Demands

Authors take part in the preparation of the final products. Not seldom, they produce most of the product.

In order to have some control, publishers can provide the tools, styles and/or strict rules.

If a separate design department is used, even not so complicated (not a designer), the author can still have some control.

Being a tool (the author's tool) is not a strength. Tools get the job done.

Authors are great at what they do.

Authors are great in print.

In service of the publisher.

User Demands

\TeX is also used by non-profit organizations, home users, those who are on their own, and those who want control (like science students).

They want access to the whole bag of tricks, in a controlled way (and in \TeX even using fonts is trickery).

Since they are willing to participate in experiments, this provides an ideal playground for new developments.

A bag of latest tricks can endanger a stable system, and in this \TeX is not different from other applications.

Apart from trickery we can try to tap the power of file formats as much as possible.

Publisher Demands

For long, \TeX was the only typesetting math, so “anything goes”. But times have changed.

More and more, graphics and layout are done in print and on desktops. What you want is what you want yourself.

We need to provide DTP capabilities, not a batch oriented program. We need the strength to provide multiple outputs.

Since “everyone can be a typographer” for design and production, the cost is relatively low (even upto the point where reuse pays off).

Step by isolated step production is not integrated workflows. The demand is for robust methods.

Since publishers want to be in business, they keep an eye on XML and \TeX can do quite well. We can even benefit from it.

We also need to educate designers in document processing: its weaknesses and

Author Demands

Authors take part in the preparation of the final products. Not seldom, they produce most of the product.

In order to have some control, publishers can provide the tools, styles and/or strict rules.

If a separate design department is used, even not so complicated (not a separate design department).

Being a tool (the tool) Tools get

Authors are great

Authors in print. In service publishers

User Demands

\TeX is also also used by non-profit organizations, home users, those who are on their own, and those who want control (like science students).

They want access to the whole bag of tricks, in a controlled way (and in \TeX even using fonts is trickery).

Since they are willing to participate in experiments, this provides an ideal playground for new developments.

A bag of latest tricks can endanger a stable system, and in this \TeX is not different from other applications.

Apart from trickery we can try to tap the power of file formats as much as possible.

Publisher Demands

For long, \TeX was the only typesetting math, so “anything but \TeX is nothing”. But times have changed.

More and more, graphics and typesetting are done in print and on desktops. What is what you want yourself.

We need to provide DTP capabilities, not just a batch oriented program. It is a strength to provide multiple

Since “everyone can be a typesetter” the demand for design and production is relatively low (even upto the reuse pays off).

Step by isolated step production is not integrated workflows. The demand robust methods.

Since publishers want to be in business, they will keep an eye on XML and \TeX can compete quite well. We can even benefit from it.

We also need to educate designers in typesetting: its weaknesses and strengths.

Author Demands

Authors take part in the preparation of the final products. Not seldom, they produce most of the product.

In order to have some control, publishers can provide the tools, styles and/or strict rules.

If a separate design department is used, even not so complicated (not a designer).

Being a typesetter is a tool (the tools get the job done).

Authors are great at design and production.

Authors are great at design and production.

In service of the publisher.

User Demands

\TeX is also used by non-profit organizations, home users, those who are on their own, and those who want control (like science students).

They want access to the whole bag of tricks, in a controlled way (and in \TeX even using fonts is trickery).

Since they are willing to participate in experiments, this provides an ideal playground for new developments.

A bag of latest tricks can endanger a stable system, and in this \TeX is not different from other applications.

Apart from trickery we can try to tap the power of file formats as much as possible.

Publisher Demands

For long, \TeX was the only typesetting math, so “anything goes”. But times have changed.

More and more, graphics and layout are done in print and on desktops. What you want is what you want yourself.

We need to provide DTP capabilities, not just a batch oriented program. We need the strength to provide multiple outputs.

Since “everyone can be a typographer” for design and production, the cost is relatively low (even upto the point where reuse pays off).

Step by isolated step production is being replaced by integrated workflows. The demand is for robust methods.

Since publishers want to be in business, they will keep an eye on XML and \TeX can compete quite well. We can even benefit from the competition.

We also need to educate designers in the use of document processing: its weaknesses and strengths.

Author Demands

Authors take part in the preparation of the final products. Not seldom, they produce most of the product.

In order to have some control, publishers can provide the tools, styles and/or strict rules.

If a separate design department is used, even not so complicated (not a designer), the author can still have a say.

Being a good tool (the author's tool) is a strength. Tools get the job done.

Authors are great. They know what they want.

Authors are great. They know what they want. They want to be in control.

In service of the publisher, the author can be a great asset.

User Demands

\TeX is also also used by non-profit organizations, home users, those who are on their own, and those who want control (like science students).

They want access to the whole bag of tricks, in a controlled way (and in \TeX even using fonts is trickery).

Since they are willing to participate in experiments, this provides an ideal playground for new developments.

A bag of latest tricks can endanger a stable system, and in this \TeX is not different from other applications.

Apart from trickery we can try to tap the power of file formats as much as possible.

Publisher Demands *Author Demands*

For long, \TeX was the only typesetting math, so “anything is possible”. But times have changed.

More and more, graphics and layout are done in print and on desktops. We

Authors take part in the preparation of the final products. Not seldom, they produce most of the product.

In order to have some control, publishers can provide the tools, styles and/or strict rules.

System Characteristics

\TeX is a batch processor as well as a language. Installation has become more easy, but the whole has become messy.

This is okay if you separate authoring and typesetting and when the design is highly structured and editors have integrated support for \TeX .

If you want fancy designs, this only pays off if you produce more than one document, or when documents change a lot. Not everyone is a hobbyist, although \TeX users can produce good output when given proper means.

*For most users, batch processing is a strange thing but this may change with faster systems. Mixed DTP and batch concepts will arise. * **

Whatever you do, if you want to reuse your information, you must have a bit of feeling for coding in the right way. Exploiting this is not trivial.

... is used, even not so complicated.

... by non-profit organizations, ... are on their own, and those (e science students).

... the whole bag of tricks, in ... in \TeX even using fonts is

... to participate in experiments, ... playground for new develop-

... s can endanger a stable sys- ... is not different from other

... ve can try to tap the power of ... as possible.

Publisher Demands *Author Demands*

For long, T_EX was the only typesetting math, so “anything is possible”. But times have changed.

More and more, graphics and layout are done in print and on desktops. We

Authors take part in the preparation of the final products. Not seldom, they produce most of the product.

In order to have some control, publishers can provide the tools, styles and/or strict rules.

System Characteristics

T_EX is a batch processor as well as a language. Installation has become more easy, but the whole has become messy.

This is okay if you separate authoring and typesetting and when the design is highly structured and editors have integrated support for T_EX.

If you want fancy designs, this only pays off if you produce more than one document, or when documents change a lot. Not everyone is a hobbyist, although T_EX users can produce good output when given proper means.

*For most users, batch processing is a strange thing but this may change with faster systems. Mixed DTP and batch concepts will arise. * **

Whatever you do, if you want to reuse your information, you must have a bit of feeling for coding in the right way. Exploiting this is not trivial.

... is used, even not so complicated.

... by non-profit organizations, ... are on their own, and those (e science students).

... the whole bag of tricks, in ... in T_EX even using fonts is

... to participate in experiments, ... playground for new develop-

... s can endanger a stable sys- ... is not different from other

... ve can try to tap the power of ... as possible.

Publisher Demands *Author Demands*

For long, \TeX was the only typesetting math, so “anything is possible”. But times have changed.

More and more, graphics and layout are done in print and on desktops. We

Authors take part in the preparation of the final products. Not seldom, they produce most of the product.

In order to have some control, publishers can provide the tools, styles and/or strict rules.

System Characteristics

\TeX is a batch processor as well as a language. Installation has become more easy, but the whole has become messy.

This is okay if you separate authoring and typesetting and when the design is highly structured and editors have integrated support for \TeX .

If you want fancy designs, this only pays off if you produce more than one document, or when documents change a lot. Not everyone is a hobbyist, although \TeX users can produce good output when given proper means.

For most users, batch processing is a strange thing but this may change with faster systems. Mixed DTP and batch concepts will arise. ✱ ✱

Whatever you do, if you want to reuse your information, you must have a bit of feeling for coding in the right way. Exploiting this is not trivial.

For the user, \TeX is used, even not so completely, for the following reasons:

• It is used by non-profit organizations, those who are on their own, and those who are not (e.g. science students).

• It is used for the whole bag of tricks, in particular in \TeX even using fonts is possible.

• It is used as a playground for new developments, to participate in experiments, to test ideas.

• It is used because it can endanger a stable system, but this is not different from other systems.

• It is used because we can try to tap the power of \TeX as possible.

Publisher Demands *Author Demands*

For long, T_EX was the only typesetting math, so “anything is possible”. But times have changed.

More and more, graphics and typesetting are done in print and on desktops. We

Authors take part in the preparation of the final products. Not seldom, they produce most of the product.

In order to have some control, publishers can provide the tools, styles and/or strict rules.

System Characteristics

T_EX is a batch processor as well as a language. Installation has become more easy, but the whole has become messy.

This is okay if you separate authoring and typesetting and when the design is highly structured and editors have integrated support for T_EX.

If you want fancy designs, this only pays off if you produce more than one document, or when documents change a lot. Not everyone is a hobbyist, although T_EX users can produce good output when given proper means.

*For most users, batch processing is a strange thing but this may change with faster systems. Mixed DTP and batch concepts will arise. * **

Whatever you do, if you want to reuse your information, you must have a bit of feeling for coding in the right way. Exploiting this is not trivial.

For the user, the system is used, even not so completely. For the publisher, the system is used, even not so completely.

by non-profit organizations, who are on their own, and those who are not (e.g. science students).

the whole bag of tricks, in the end, in T_EX even using fonts is possible.

to participate in experiments, and to have a playground for new developments.

s can endanger a stable system. T_EX is not different from other systems.

we can try to tap the power of the system as possible.

Publisher Demands *Author Demands*

For long, T_EX was the only typesetting math, so “anything is possible”. But times have changed.

More and more, graphics and layout are done in print and on desktops. We

Authors take part in the preparation of the final products. Not seldom, they produce most of the product.

In order to have some control, publishers can provide the tools, styles and/or strict rules.

System Characteristics

T_EX is a batch processor as well as a language. Installation has become more easy, but the whole has become messy.

This is okay if you separate authoring and typesetting and when the design is highly structured and editors have integrated support for T_EX.

If you want fancy designs, this only pays off if you produce more than one document, or when documents change a lot. Not everyone is a hobbyist, although T_EX users can produce good output when given proper means.

For most users, batch processing is a strange thing but this may change with faster systems. Mixed DTP and batch concepts will arise. ★ ★

Whatever you do, if you want to reuse your information, you must have a bit of feeling for coding in the right way. Exploiting this is not trivial.

For the publisher, T_EX is used, even not so completely, for the following reasons:

• It is easy to use, even for non-professionals.

• It is supported by non-profit organizations, so users are on their own, and those who are not (e.g. science students).

• It is possible to have the whole bag of tricks, in particular, to do it in T_EX even using fonts is possible.

• It is possible to participate in experiments, and to have a playground for new developments.

• It is possible to endanger a stable system, but this is not different from other systems.

• It is possible to try to tap the power of the system as far as possible.

Publisher Demands *Author Demands*

For long, T_EX was the only typesetting math, so “anything is possible”. But times have changed.

More and more, graphics and layout are done in print and on desktops. We

Authors take part in the preparation of the final products. Not seldom, they produce most of the product.

In order to have some control, publishers can provide the tools, styles and/or strict rules.

System Characteristics

T_EX is a batch processor as well as a language. Installation has become more easy, but the whole has become messy.

This is okay if you separate authoring and typesetting and when the design is highly structured and editors have integrated support for T_EX.

If you want fancy designs, this only pays off if you produce more than one document, or when documents change a lot. Not everyone is a hobbyist, although T_EX users can produce good output when given proper means.

For most users, batch processing is a strange thing but this may change with faster systems. Mixed DTP and batch concepts will arise. ★ ★

Whatever you do, if you want to reuse your information, you must have a bit of feeling for coding in the right way. Exploiting this is not trivial.

For the user, the system is used, even not so completely. For the publisher, the system is used, even not so completely.

by non-profit organizations, who are on their own, and those who are not (e.g. science students).

the whole bag of tricks, in the end, in T_EX even using fonts is possible.

to participate in experiments, and to have a playground for new developments.

s can endanger a stable system. T_EX is not different from other systems.

we can try to tap the power of the system as possible.

Turning Tides

With the advance of highly structured formats, like XML, more verbose code gains attentions. In \TeX we can use mixed those approaches.

Since such formats pose strong limitations, for \TeX life has become easier.

A pitfall is that the problems (i.e. automated processing) have hardly changed, at least not for high-end products.

If we want original products, we still need flexible systems. In this respect \TeX is still a good choice.

In many cases, flexibility and/or original solutions come not out-of-the-box (yet).

Due to far better PR, XML and related tools gain much more attention than \TeX , and we can learn from that.

For most users, batch processing is a strange thing but this may change with faster systems. Mixed DTP and batch concepts will arise. ✱ ✱

Whatever you do, if you want to reuse your information, you must have a bit of feeling for coding in the right way. Exploiting this is not trivial.

... the preparation of the final products. ... most of the product.

... control, publishers can provide the tools, ...

... t is used, even not so compli- ...

... ation ...

... by non-profit organizations, ... are on their own, and those (e science students).

... the whole bag of tricks, in d in \TeX even using fonts is

... to produce a lot. ... to produce ... to participate in experiments, ... playground for new develop-

... s can endanger a stable sys- ... is not different from other

... we can try to tap the power of ... as possible.

Turning Tides

With the advance of highly structured formats, like XML, more verbose code gains attentions. In \TeX we can use mix those approaches.

Since such formats pose strong limitations, for \TeX life has become easier.

A pitfall is that the problems (i.e. automated processing) have hardly changed, at least not for high-end products.

If we want original products, we still need flexible systems. In this respect \TeX is still a good choice.

In many cases, flexibility and/or original solutions come not out-of-the-box (yet).

Due to far better PR, XML and related tools gain much more attention than \TeX , and we can learn from that.

For most users, batch processing is a strange thing but this may change with faster systems. Mixed DTP and batch concepts will arise. ✱ ✱

Whatever you do, if you want to reuse your information, you must have a bit of feeling for coding in the right way. Exploiting this is not trivial.

the preparation of the final products. The most of the product.

control, publishers can provide the tools,

it is used, even not so compli-

ation

by non-profit organizations, who are on their own, and those (e science students).

ing and
inte-

the whole bag of tricks, in addition in \TeX even using fonts is

produce

a lot.

produce

to participate in experiments, playground for new develop-

s can endanger a stable system. It is not different from other

we can try to tap the power of as possible.

Turning Tides

With the advance of highly structured formats, like XML, more verbose code gains attentions. In \TeX we can use mix those approaches.

Since such formats pose strong limitations, for \TeX life has become easier.

A pitfall is that the problems (i.e. automated processing) have hardly changed, at least not for high-end products.

If we want original products, we still need flexible systems. In this respect \TeX is still a good choice.

In many cases, flexibility and/or original solutions come not out-of-the-box (yet).

Due to far better PR, XML and related tools gain much more attention than \TeX , and we can learn from that.

For most users, batch processing is a strange thing but this may change with faster systems. Mixed DTP and batch concepts will arise. ✱ ✱

Whatever you do, if you want to reuse your information, you must have a bit of feeling for coding in the right way. Exploiting this is not trivial.

the preparation of the final products. The most of the product.

control, publishers can provide the tools,

it is used, even not so compli-

ation

by non-profit organizations, who are on their own, and those (e science students).

ing and
inte-

the whole bag of tricks, in addition in \TeX even using fonts is

produce

a lot.

produce

to participate in experiments, playground for new develop-

s can endanger a stable system. It is not different from other

we can try to tap the power of as possible.

Turning Tides

With the advance of highly structured formats, like XML, more verbose code gains attentions. In \TeX we can use mix those approaches.

Since such formats pose strong limitations, for \TeX life has become easier.

A pitfall is that the problems (i.e. automated processing) have hardly changed, at least not for high-end products.

If we want original products, we still need flexible systems. In this respect \TeX is still a good choice.

In many cases, flexibility and/or original solutions come not out-of-the-box (yet).

Due to far better PR, XML and related tools gain much more attention than \TeX , and we can learn from that.

For most users, batch processing is a strange thing but this may change with faster systems. Mixed DTP and batch concepts will arise. ✱ ✱

Whatever you do, if you want to reuse your information, you must have a bit of feeling for coding in the right way. Exploiting this is not trivial.

the preparation of the final products. The most of the product.

control, publishers can provide the tools,

it is used, even not so compli-

ation

by non-profit organizations, who are on their own, and those (e science students).

7 and
inte-

the whole bag of tricks, in d in \TeX even using fonts is

produce

a lot.

produce

to participate in experiments, playground for new develop-

s can endanger a stable sys- C is not different from other

we can try to tap the power of as possible.

Turning Tides

With the advance of highly structured formats, like XML, more verbose code gains attentions. In \TeX we can use mix those approaches.

Since such formats pose strong limitations, for \TeX life has become easier.

A pitfall is that the problems (i.e. automated processing) have hardly changed, at least not for high-end products.

If we want original products, we still need flexible systems. In this respect \TeX is still a good choice.

In many cases, flexibility and/or original solutions come not out-of-the-box (yet).

Due to far better PR, XML and related tools gain much more attention than \TeX , and we can learn from that.

For most users, batch processing is a strange thing but this may change with faster systems. Mixed DTP and batch concepts will arise. ✱ ✱

Whatever you do, if you want to reuse your information, you must have a bit of feeling for coding in the right way. Exploiting this is not trivial.

the preparation of the final products. The most of the product.

control, publishers can provide the tools,

it is used, even not so compli-

ation

by non-profit organizations, who are on their own, and those (e science students).

7 and
inte-

the whole bag of tricks, in d in \TeX even using fonts is

produce

a lot.

produce

to participate in experiments, playground for new develop-

s can endanger a stable sys- C is not different from other

we can try to tap the power of as possible.

Turning Tides

With the advance of highly structured formats, like XML, more verbose code gains attentions. In \TeX we can use mix those approaches.

Since such formats pose strong limitations, for \TeX life has become easier.

A pitfall is that the problems (i.e. automated processing) have hardly changed, at least not for high-end products.

If we want original products, we still need flexible systems. In this respect \TeX is still a good choice.

In many cases, flexibility and/or original solutions come not out-of-the-box (yet).

Due to far better PR, XML and related tools gain much more attention than \TeX , and we can learn from that.

For most users, batch processing is a strange thing but this may change with faster systems. Mixed DTP and batch concepts will arise. ✱ ✱

Whatever you do, if you want to reuse your information, you must have a bit of feeling for coding in the right way. Exploiting this is not trivial.

the preparation of the final products. The most of the product.

control, publishers can provide the tools,

it is used, even not so compli-

ation

by non-profit organizations, who are on their own, and those (e science students).

7 and
inte-

the whole bag of tricks, in addition in \TeX even using fonts is

produce

a lot.

produce

to participate in experiments, playground for new develop-

s can endanger a stable system. It is not different from other

we can try to tap the power of as possible.

Turning Tides

With the advance of highly structured formats, like XML, more verbose code gains attentions. In \TeX we can use mixed those approaches.

Since such formats pose strong limitations, for \TeX life has become easier.

A pitfall is that the problems (i.e. automated processing) have hardly changed, at least not for high-end products.

If we want original products, we still need flexible systems. In this respect \TeX is still a good choice.

In many cases, flexibility and/or original solutions come not out-of-the-box (yet).

Due to far better PR, XML and related tools gain much more attention than \TeX , and we can learn from that.

For most users, batch processing is a strange thing but this may change with faster systems. Mixed DTP and batch concepts will arise. ✱ ✱

Whatever you do, if you want to reuse your information, you must have a bit of feeling for coding in the right way. Exploiting this is not trivial.

the preparation of the final products. The most of the product.

control, publishers can provide the tools,

it is used, even not so compli-

ation

by non-profit organizations, who are on their own, and those (e science students).

7 and
inte-

the whole bag of tricks, in addition in \TeX even using fonts is

produce

a lot.

produce

to participate in experiments, playground for new develop-

s can endanger a stable system. It is not different from other

we can try to tap the power of as possible.

Turning Tides

With the advance of highly structured formats, like XML, more verbose code gains attentions. In \TeX we can use mix those approaches.

Since such formats pose strong limitations, for \TeX life has become easier.

A pitfall is that the problems (i.e. automated processing) have hardly changed, at least not for high-end products.

If we want systems. In

In many ca come not ou

Due to far better PR, XML and related tools gain much more attention than \TeX , and we can learn from that.

For most users, batch processing is a strange thing but this may change with faster systems. Mixed DTP and batch concepts will arise. ☆☆

Whatever you do, if you want to reuse your information, you must have a bit of feeling for coding in the right way. Exploiting this is not trivial.

\TeX in 2001:

we're still not outdated.

s
the preparation of the final products.
e most of the product.

ontrol, publishers can provide the tools,

t is used, even not so compli-

ation

rganizations,
n, and those
s).

of tricks, in
sing fonts is

roduce

a lot.

roduce

to participate in experiments,
playground for new develop-

s can endanger a stable sys-
C is not different from other

we can try to tap the power of
as possible.