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.

TEX 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.

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.

TEX 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.

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.

TEX 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.

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.

TEX 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.

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.

TEX 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.

TEX in 2001

are we still up-to-date?

Competing with DTP

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

Fonts, spacing, color and all relevan
be influenced.

TEX is a potential winner when it
graphics but designers don't think t

Desk top application come with good
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
anywhere on the page.

Fonts, spacing, color and all relevan
be influenced.

TEX is a potential winner when it
graphics but designers don't think t

Desk top application come with good
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
anywhere on the page.

Fonts, spacing, color and all relevan
be influenced.

TEX is a potential winner when it
graphics but designers don't think t

Desk top application come with good
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
anywhere on the page.

Fonts, spacing, color and all relevan
be influenced.

TEX is a potential winner when it
graphics but designers don't think t

Desk top application come with good
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
anywhere on the page.

Fonts, spacing, color and all relevan
be influenced.

TEX is a potential winner when it
graphics but designers don't think t

Desk top application come with good
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
anywhere on the page.

Fonts, spacing, color and all relevan
be influenced.

TEX is a potential winner when it
graphics but designers don't think t

Desk top application come with good
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
anywhere on the page.

Fonts, spacing, color and all relevan
be influenced.

TEX is a potential winner when it
graphics but designers don't think t

Desk top application come with good
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
anywhere on the page.

Fonts, spacing, color and all relevan
be influenced.

TEX is a potential winner when it
graphics but designers don't think t

Desk top application come with good
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
anywhere on the page.

Fonts, spacing, color and all relevan
be influenced.

TEX is a potential winner when it
graphics but designers don't think t

Desk top application come with good
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
anywhere on the page.

Fonts, spacing, color and all relevan
be influenced.

TEX is a potential winner when it
graphics but designers don't think t

Desk top application come with good
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
anywhere on the page.

Fonts, spacing, color and all relevan
be influenced.

TEX is a potential winner when it
graphics but designers don't think t

Desk top application come with good
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.

rst look slow, demanding, old fash-
ntroduced and supported properly
ympathy.

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 righ 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 strength.

uthoring

ian alone text, so we need to provide

logy used, such elements are fragile
e.

eed to guard the traditional means

rther than any other system, simply
.

Competing with DTP

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

Fonts, spacing, color and all relevan
be influenced.

TEX is a potential winner when it
graphics but designers don't think t

Desk top application come with good
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.

rst look slow, demanding, old fash-
ntroduced and supported properly
ympathy.

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 righ 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 strength.

uthoring

ian alone text, so we need to provide

logy used, such elements are fragile
e.

eed to guard the traditional means

rther than any other system, simply
.

Competing with DTP

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

Fonts, spacing, color and all relevan
be influenced.

TEX is a potential winner when it
graphics but designers don't think t

Desk top application come with good
courses and support

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 righ 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 strength.

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.

rst look slow, demanding, old fash-
ntroduced and supported properly
ympathy.

luthoring

ian alone text, so we need to provide

logy used, such elements are fragile
e.

eed to guard the traditional means

rther than any other system, simply
.

Competing with DTP

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

Fonts, spacing, color and all relevan
be influenced.

TEX is a potential winner when it
graphics but designers don't think t

Desk top application come with good
courses and support.

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 righ 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 strength.

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.

rst look slow, demanding, old fash-
ntroduced and supported properly
ympathy.

luthoring

ian alone text, so we need to provide

logy used, such elements are fragile
e.

eed to guard the traditional means

rther than any other system, simply
.

Competing with DTP

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

Fonts, spacing, color and all relevan
be influenced.

TEX is a potential winner when it
graphics but designers don't think t

Desk top application come with good
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.

rst look slow, demanding, old fash-
ntroduced and supported properly
ympathy.

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 righ 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 strength.

uthoring

ian alone text, so we need to provide

ogy used, such elements are fragile
e.

eed to guard the traditional means

rther than any other system, simply
.

Competing with DTP

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

Fonts, spacing, color and all relevan
be influenced.

TEX is a potential winner when it
graphics but designers don't think t

Desk top application come with good
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.

rst look slow, demanding, old fash-
ntroduced and supported properly
ympathy.

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 righ 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 strength.

luthoring

ian alone text, so we need to provide

logy used, such elements are fragile
e.

eed to guard the traditional means

rther 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.

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 strength.

mitations

you need to catch a lot of border

lity, you also provide an opening
d (in case of a reprogrammable

trol is wanted, the impossible is
So, providing structured control
y.

look slow, demanding, old fash-
roduced and supported properly
pathy.

thoring

alone text, so we need to provide

y used, such elements are fragile

ed to guard the traditional means

rther 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.

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 strength.

mitations

you need to catch a lot of border

lity, you also provide an opening
d (in case of a reprogrammable

trol is wanted, the impossible is
So, providing structured control
y.

look slow, demanding, old fash-
roduced and supported properly
pathy.

thoring

alone text, so we need to provide

y used, such elements are fragile

ed to guard the traditional means

rther 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.

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 strength.

mitations

you need to catch a lot of border

lity, you also provide an opening
d (in case of a reprogrammable

trol is wanted, the impossible is
So, providing structured control
y.

look slow, demanding, old fash-
roduced and supported properly
pathy.

thoring

alone text, so we need to provide

y used, such elements are fragile

ed to guard the traditional means

rther 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.

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 strength.

mitations

you need to catch a lot of border

lity, you also provide an opening
d (in case of a reprogrammable

trol is wanted, the impossible is
So, providing structured control
y.

look slow, demanding, old fash-
roduced and supported properly
pathy.

thoring

alone text, so we need to provide

y used, such elements are fragile

ed to guard the traditional means

rther 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.

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 strength.

mitations

you need to catch a lot of border

lity, you also provide an opening
d (in case of a reprogrammable

trol is wanted, the impossible is
So, providing structured control
y.

look slow, demanding, old fash-
roduced and supported properly
pathy.

thoring

alone text, so we need to provide

y used, such elements are fragile

ed to guard the traditional means

rther 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.

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 strength.

mitations

you need to catch a lot of border

lity, you also provide an opening
d (in case of a reprogrammable

trol is wanted, the impossible is
So, providing structured control
y.

look slow, demanding, old fash-
roduced and supported properly
pathy.

thoring

alone text, so we need to provide

y used, such elements are fragile

ed to guard the traditional means

rther 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.

(decent) things that designers want to do.

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

mitations

you need to catch a lot of border

lity, you also provide an opening
d (in case of a reprogrammable

trol is wanted, the impossible is
So, providing structured control
y.

look slow, demanding, old fash-
roduced and supported properly
pathy.

thoring

alone text, so we need to provide

y used, such elements are fragile

ed to guard the traditional means

rther than any other system, simply

Publisher Demands

For long, TEX was the only affor
ting math, so "anything was be
times have changed.

More and more, graphics and c
print an on desktops. What yo
you want yourself.

We need to provide DTP compet
oriented program. Here we ca
provide multiple products.

Since "everyone can be a types
design and production of docur
(even upto the unacceptable). Bu

Step by isolated step production
workflows. The often tight sc
methods.

Since publishers want to be in bu
eye on XML and TEX can hand
can even benefit from it.

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.

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 strength.

ed to guard the traditional means

rther than any other system, simply

Publisher Demands

For long, TEX was the only affor
ting math, so "anything was be
times have changed.

More and more, graphics and c
print an on desktops. What yo
you want yourself.

We need to provide DTP compet
oriented program. Here we ca
provide multiple products.

Since "everyone can be a types
design and production of docur
(even upto the unacceptable). Bu

Step by isolated step production
workflows. The often tight sc
methods.

Since publishers want to be in bu
eye on XML and TEX can hand
can even benefit from it.

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.

ed to guard the traditional means

(decent) things that designers want to do.

We also need to educate designers in automatic docu-
ment processing: its weaknesses and strength.

rther than any other system, simply

Publisher Demands

For long, TEX was the only affor
ting math, so "anything was be
times have changed.

More and more, graphics and c
print an on desktops. What yo
you want yourself.

We need to provide DTP compet
oriented program. Here we ca
provide multiple products.

Since "everyone can be a types
design and production of docur
(even upto the unacceptable). Bu

Step by isolated step production
workflows. The often tight sc
methods.

Since publishers want to be in bu
eye on XML and TEX can hand
can even benefit from it.

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.

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 strength.

ed to guard the traditional means

rther than any other system, simply

Publisher Demands

For long, TEX was the only affor
ting math, so "anything was be
times have changed.

More and more, graphics and c
print an on desktops. What yo
you want yourself.

We need to provide DTP compet
oriented program. Here we ca
provide multiple products.

Since "everyone can be a types
design and production of docur
(even upto the unacceptable). Bu

Step by isolated step production
workflows. The often tight sc
methods.

Since publishers want to be in bu
eye on XML and TEX can hand
can even benefit from it.

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.

ed to guard the traditional means

(decent) things that designers want to do.

We also need to educate designers in automatic docu-
ment processing: its weaknesses and strength.

rther than any other system, simply

Publisher Demands

For long, TEX was the only affor
ting math, so "anything was be
times have changed.

More and more, graphics and c
print an on desktops. What yo
you want yourself.

We need to provide DTP compet
oriented program. Here we ca
provide multiple products.

Since "everyone can be a types
design and production of docur
(even upto the unacceptable). Bu

Step by isolated step production
workflows. The often tight sc
methods.

Since publishers want to be in bu
eye on XML and TEX can hand
can even benefit from it.

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.

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 strength.

ed to guard the traditional means

rther than any other system, simply

Publisher Demands

For long, TEX was the only affor
ting math, so "anything was be
times have changed.

More and more, graphics and c
print an on desktops. What yo
you want yourself.

We need to provide DTP compet
oriented program. Here we ca
provide multiple products.

Since "everyone can be a types
design and production of docur
(even upto the unacceptable). Bu

Step by isolated step production
workflows. The often tight sc
methods.

Since publishers want to be in bu
eye on XML and TEX can hand
can even benefit from it.

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.

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 strength.

ed to guard the traditional means

rther than any other system, simply

Publisher Demands

For long, TEX was the only affor
ting math, so "anything was be
times have changed.

More and more, graphics and c
print an on desktops. What yo
you want yourself.

We need to provide DTP compet
oriented program. Here we ca
provide multiple products.

Since "everyone can be a types
design and production of docur
(even upto the unacceptable). Bu

Step by isolated step production
workflows. The often tight sc
methods.

Since publishers want to be in bu
eye on XML and TEX can hand
can even benefit from it.

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.

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 strength.

ed to guard the traditional means

rther than any other system, simply

Publisher Demands

For long, TEX was the only affor
ting math, so "anything was be
times have changed.

More and more, graphics and c
print an on desktops. What yo
you want yourself.

We need to provide DTP compet
oriented program. Here we ca
provide multiple products.

Since "everyone can be a types
design and production of docur
(even upto the unacceptable). Bu

Step by isolated step production
workflows. The often tight sc
methods.

Since publishers want to be in bu
eye on XML and TEX can hand
can even benefit from it.

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.

ed to guard the traditional means

(decent) things that designers want to do.

We also need to educate designers in automatic docu-
ment processing: its weaknesses and strength.

rther than any other system, simply

Publisher Demands

For long, TEX was the only affo
ting math, so "anything was be
times have changed.

More and more, graphics and c
print an on desktops. What yo
you want yourself.

We need to provide DTP compet
oriented program. Here we ca
provide multiple products.

Since "everyone can be a types
design and production of docu
(even upto the unacceptable). Bu

Step by isolated step production
workflows. The often tight sc
methods.

Since publishers want to be in bu
eye on XML and TEX can hand
can even benefit from it.

We have to make sure that TEX can
(decent) things that designers want to d

We also need to educate designers in au
ment processing: its weaknesses and stre

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
compl
loops.

Being
good
appli

Auth
result

Auth
see in

In ser
of pu

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 affo
ting math, so "anything was be
times have changed.

More and more, graphics and c
print an on desktops. What yo
you want yourself.

We need to provide DTP compet
oriented program. Here we ca
provide multiple products.

Since "everyone can be a types
design and production of docur
(even upto the unacceptable). Bu

Step by isolated step production
workflows. The often tight sc
methods.

Since publishers want to be in bu
eye on XML and TEX can hand
can even benefit from it.

We have to make sure that TEX can
(decent) things that designers want to d

We also need to educate designers in au
ment processing: its weaknesses and stre

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 compl
loops.

Being
good
appli

Auth
result

Auth
see in

In ser
of pu

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 affo
ting math, so "anything was be
times have changed.

More and more, graphics and c
print an on desktops. What yo
you want yourself.

We need to provide DTP compet
oriented program. Here we ca
provide multiple products.

Since "everyone can be a types
design and production of docur
(even upto the unacceptable). Bu

Step by isolated step production
workflows. The often tight sc
methods.

Since publishers want to be in bu
eye on XML and TEX can hand
can even benefit from it.

(decent) things that designers want to d

We also need to educate designers in au
ment processing: its weaknesses and stre

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
loops.

Being
good
appli

Auth
result

Auth
see in

In ser
of pu

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 affo
ting math, so "anything was be
times have changed.

More and more, graphics and c
print an on desktops. What yo
you want yourself.

We need to provide DTP compet
oriented program. Here we ca
provide multiple products.

Since "everyone can be a types
design and production of docur
(even upto the unacceptable). Bu

Step by isolated step production
workflows. The often tight sc
methods.

Since publishers want to be in bu
eye on XML and TEX can hand
can even benefit from it.

(decent) things that designers want to d

We also need to educate designers in au
ment processing: its weaknesses and stre

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
loops.

Being
good
appli

Auth
result

Auth
see in

In ser
of pu

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 affo
ting math, so "anything was be
times have changed.

More and more, graphics and c
print an on desktops. What yo
you want yourself.

We need to provide DTP compet
oriented program. Here we ca
provide multiple products.

Since "everyone can be a types
design and production of docu
(even upto the unacceptable). Bu

Step by isolated step production
workflows. The often tight sc
methods.

Since publishers want to be in bu
eye on XML and TEX can hand
can even benefit from it.

(decent) things that designers want to d

We also need to educate designers in au
ment processing: its weaknesses and stre

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
loops.

Being
good
appli

Auth
result

Auth
see in

In se
of pu

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 affo
ting math, so "anything was be
times have changed.

More and more, graphics and c
print an on desktops. What yo
you want yourself.

We need to provide DTP compet
oriented program. Here we ca
provide multiple products.

Since "everyone can be a types
design and production of docur
(even upto the unacceptable). Bu

Step by isolated step production
workflows. The often tight sc
methods.

Since publishers want to be in bu
eye on XML and TEX can hand
can even benefit from it.

We have to make sure that TEX can
(decent) things that designers want to d

We also need to educate designers in au
ment processing: its weaknesses and stre

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 compl
loops.

Being
good
appli

Auth
result

Auth
see in

In ser
of pu

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 affo
ting math, so "anything was be
times have changed.

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
ict rules.

department is used, even not so

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 of 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.

nands

o used by non-profit organizations,
ose who are on their own, and those
trol (like science students).

cess to the whole bag of tricks, in
ay (and in TEX even using fonts is

willing to participate in experi-
ovides an ideal playground for new

t tricks can endanger a stable sys-
is TEX is not different from other

ickery we can try to tap the power
as much as possible.

Publisher Demands

For long, TEX was the only affo
ting math, so "anything was be
times have changed.

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
ict rules.

department is used, even not so

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 of 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.

nands

o used by non-profit organizations,
ose who are on their own, and those
trol (like science students).

cess to the whole bag of tricks, in
ay (and in TEX even using fonts is

willing to participate in experi-
ovides an ideal playground for new

t tricks can endanger a stable sys-
is TEX is not different from other

ickery we can try to tap the power
as much as possible.

Publisher Demands

For long, TEX was the only affo
ting math, so "anything was be
times have changed.

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
ict rules.

department is used, even not so

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 of 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.

nands

o used by non-profit organizations,
ose who are on their own, and those
trol (like science students).

cess to the whole bag of tricks, in
ay (and in TEX even using fonts is

willing to participate in experi-
ovides an ideal playground for new

t tricks can endanger a stable sys-
is TEX is not different from other

ickery we can try to tap the power
as much as possible.

Publisher Demands

For long, TEX was the only affo
ting math, so "anything was be
times have changed.

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
ict rules.

department is used, even not so

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 of 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.

nands

o used by non-profit organizations,
ose who are on their own, and those
trol (like science students).

cess to the whole bag of tricks, in
ay (and in TEX even using fonts is

willing to participate in experi-
ovides an ideal playground for new

t tricks can endanger a stable sys-
is TEX is not different from other

ickery we can try to tap the power
as much as possible.

Publisher Demands

For long, TEX was the only affo
ting math, so "anything was be
times have changed.

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
ict rules.

department is used, even not so

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 of 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.

nands

o used by non-profit organizations,
ose who are on their own, and those
trol (like science students).

cess to the whole bag of tricks, in
ay (and in TEX even using fonts is

willing to participate in experi-
ovides an ideal playground for new

t tricks can endanger a stable sys-
is TEX is not different from other

ickery we can try to tap the power
as much as possible.

Publisher Demands

For long, TEX was the only affo
ting math, so "anything was be
times have changed.

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
ict rules.

department is used, even not so

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 of 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.

nands

o used by non-profit organizations,
ose who are on their own, and those
trol (like science students).

cess to the whole bag of tricks, in
ay (and in TEX even using fonts is

willing to participate in experi-
ovides an ideal playground for new

t tricks can endanger a stable sys-
is TEX is not different from other

ickery we can try to tap the power
as much as possible.

Turning Tides

With the advance of highly structured formats, like XML, more verbose code gains attentions. In TEX we can used 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.

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.

ls

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

ontrol, publishers can provide the
ct rules.

epartment is used, even not so

ands

used by non-profit organizations,
se who are on their own, and those
rol (like science students).

ess to the whole bag of tricks, in
y (and in TEX even using fonts is

willing to participate in experi-
vides an ideal playground for new

t tricks can endanger a stable sys-
is TEX is not different from other

ickery we can try to tap the power
as much as possible.

Turning Tides

With the advance of highly structured formats, like XML, more verbose code gains attentions. In TEX we can used 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.

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.

ls

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

ontrol, publishers can provide the
ct rules.

epartment is used, even not so

ands

used by non-profit organizations,
se who are on their own, and those
trol (like science students).

ess to the whole bag of tricks, in
y (and in TEX even using fonts is

willing to participate in experi-
vides an ideal playground for new

t tricks can endanger a stable sys-
is TEX is not different from other

ickery we can try to tap the power
as much as possible.

Turning Tides

With the advance of highly structured formats, like XML, more verbose code gains attentions. In TEX we can used 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.

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.

ls

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

ontrol, publishers can provide the
ct rules.

epartment is used, even not so

ands

used by non-profit organizations,
se who are on their own, and those
trol (like science students).

ess to the whole bag of tricks, in
y (and in TEX even using fonts is

willing to participate in experi-
vides an ideal playground for new

t tricks can endanger a stable sys-
is TEX is not different from other

ickery we can try to tap the power
as much as possible.

Turning Tides

With the advance of highly structured formats, like XML, more verbose code gains attentions. In TEX we can used 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.

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.

ls

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

ontrol, publishers can provide the
ct rules.

epartment is used, even not so

ands

used by non-profit organizations,
se who are on their own, and those
trol (like science students).

ess to the whole bag of tricks, in
y (and in TEX even using fonts is

willing to participate in experi-
vides an ideal playground for new

t tricks can endanger a stable sys-
is TEX is not different from other

ickery we can try to tap the power
as much as possible.

Turning Tides

With the advance of highly structured formats, like XML, more verbose code gains attentions. In TEX we can used 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.

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.

ls

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

ontrol, publishers can provide the
ct rules.

epartment is used, even not so

ands

used by non-profit organizations,
se who are on their own, and those
trol (like science students).

ess to the whole bag of tricks, in
y (and in TEX even using fonts is

willing to participate in experi-
vides an ideal playground for new

t tricks can endanger a stable sys-
is TEX is not different from other

ickery we can try to tap the power
as much as possible.

Turning Tides

With the advance of highly structured formats, like XML, more verbose code gains attentions. In TEX we can used 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.

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.

ls

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

ontrol, publishers can provide the
ct rules.

epartment is used, even not so

ands

used by non-profit organizations,
se who are on their own, and those
rol (like science students).

ess to the whole bag of tricks, in
y (and in TEX even using fonts is

willing to participate in experi-
vides an ideal playground for new

t tricks can endanger a stable sys-
is TEX is not different from other

ickery we can try to tap the power
as much as possible.

Turning Tides

With the advance of highly structured formats, like XML, more verbose code gains attentions. In TEX we can used 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.

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.

ls

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

ontrol, publishers can provide the
ct rules.

epartment is used, even not so

ands

used by non-profit organizations,
se who are on their own, and those
trol (like science students).

ess to the whole bag of tricks, in
y (and in TEX even using fonts is

willing to participate in experi-
ovides an ideal playground for new

t tricks can endanger a stable sys-
is TEX is not different from other

ickery we can try to tap the power
as much as possible.

Turning Tides

With the advance of highly structured formats, like XML, more verbose code gains attentions. In TEX we can used 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 wan… systems. In …

In many cas… not out-of-t…

Due to far better LaTeX/XML and related tools gain much more attention than TEX, and we can learn from that.

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.

…ls

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

…ontrol, publishers can provide the …ct rules.

…epartment is used, even not so …

…ands

…ganizations, …m, and those …ents).

…of tricks, in …sing fonts is …

…willing to participate in experi- …ovides an ideal playground for new …

…t tricks can endanger a stable sys- …is TEX is not different from other …

…ickery we can try to tap the power … as much as possible.

TEX in 2001

we're still not outdated.