

7363 - Web-basierte Anwendungen

Eine Vertiefungsveranstaltung
mit Schwerpunkt auf XML-Technologien

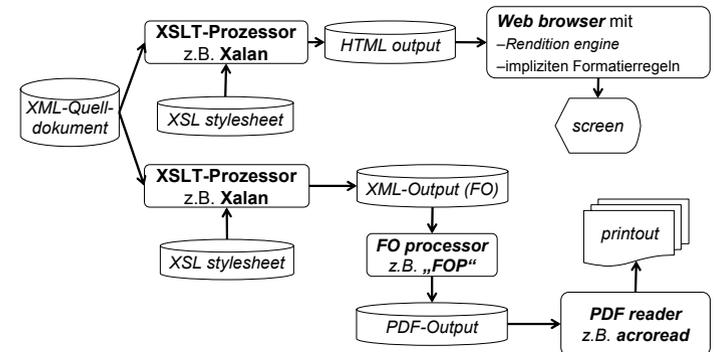
XSL-FO

XSL Formatting Objects

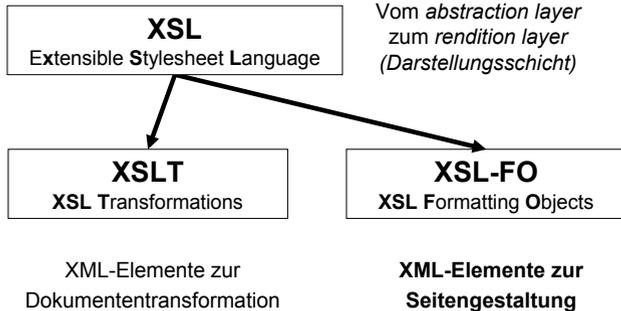
Einführung

XSL-FO: Ablaufschema

Erinnerung:



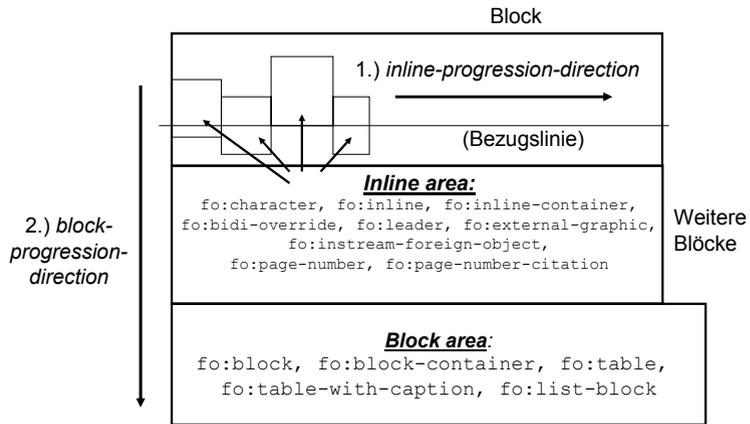
Erinnerung:



- Status:
 - W3C *Recommendation* 1.0, 15.10.2001
 - Quelle: <http://www.w3.org/TR/xsl/>
 - W3C *Candidate Recommendation* (CR) 1.1, 20.02.2006
 - Quelle: <http://www.w3.org/TR/xsl11/>
- XML-FO bildet wohlgeformte XML-Dokumente mit Namensraumunterstützung
 - Name des Dokumentenelements:
 - `root`
 - Namensraum URI:
 - `http://www.w3.org/1999/XSL/Format`
 - Konventionen:
 - Datenendungen: `*.fo`, `*.xml` (warum?)
 - Präfix: `fo:`

- Quellen:
 - [1] <http://www.w3.org/TR/xsl/>
 - Die Spezifikation, 8 Kapitel + 7 Anhänge
 - [2] <http://www.ibiblio.org/xml/books/bible2/chapters/ch18.html>
 - Kapitel 18 des Buchs "The XML Bible 2nd ed.": *XSL Formatting Objects* (versteht sich als kostenlose Leseprobe)
 - Gut verständlich, systematisch, enthält das Wichtigste
 - [3] <http://www.renderx.com/tutorial.html>
 - Ein leicht verständliches Tutorial, von Mitwirkenden am Standard
 - Werbung für ein XSL Formatter-Produkt "XEP", doch vom W3C genannt
 - [4] *XSL-FO. Making XML Look Good in Print.* Dave Pawson, O'Reilly, Sebastopol, CA, 2002. ISBN: 0-596-00355-2
 - Zur Zeit das Standardwerk zu XSL-FO

- Kernkonzepte
 - **Seitenorientierte Ausgabe**
 - Nicht auf Papierdruck beschränkt, aber klar im Vordergrund
 - Ausgabe z.B. in Browser denkbar ("eine sehr lange Seite")
 - Seitenvorlagen ("*page masters*") vs. Inhalt
 - Konzept vergleichbar mit MS PowerPoint's "*page masters*"
 - Statische vs. variable Inhalte
 - Wiederkehrende Ausgaben wie Seitennummern, Buch- oder Kapitel-Titel, Logos, Hintergrundgestaltung werden separat vom eigentlichen - variablen - Inhalt verwaltet.
 - **Blöcke und *Inline*-Elemente**
 - Ausgabegebiete sind i.w. **Rechtecke**
 - Gestaltung der Rechtecke und deren Begrenzungen ist wesentlich
 - *Inline*-Gestaltung: Ähnlich wie bei CSS, Erinnerung an DocBook



• Kernkonzepte

– Sprachunabhängigkeit

- XSL-FO ist auch für nicht-westliche Sprachen geeignet, die unseren gewohnten Lesefluss (links nach rechts, oben nach unten; engl.: **lr-tb**) nicht aufweisen.
- Hebräisch, Arabisch: Rechts nach links, oben nach unten (**rl-tb**)
- Japanisch etc.: Oben nach unten, rechts nach links (**tb-rl**)
- Auch bidirektionale u.a. Varianten!

– Konsequenzen

- Abstraktere Positions- und Reihenfolgeangaben als die "üblichen" wurden ergänzt
- Sprachschlüssel erforderlich (zur Steuerung des FO-Prozessors)
- Verwechslungsgefahr wegen Redundanz bei einigen Attributen

• Kernkonzepte

– Deklarative Formulierungen

- Der FO-Prozessor entscheidet z.B. selbsttätig über Zeilen- und Seitenumbrüche, Silbentrennung etc.
- Anwender können beim FO-Prozessor „Wünsche anmelden“
 - z.B. Seitenumbruch „empfehlen“ oder die Zahl der "widows & orphans" (Druckersprache) verändern
- Bei Konflikten zwischen Anforderungen (!) entscheidet der Prozessor

– Integrierbare Grafik

- Fremdgrafiken lassen sich importieren,
- SVG-Grafiken lassen sich einbetten

– Eigenes Tabellenmodell

• Technische Parallelen

– CSS

- Viele Gemeinsamkeiten, Co-Entwicklung
- Namen von FO-Attributen sind oft identisch mit CSS-Bezeichnern.

– DSSSL

- Das Vorbild aus SGML
- Siehe auch das CSS-Kapitel zur XML-LV

– TeX (sprich: "tech", X = griech. "Chi", also etwa: Te χ)

- Hochentwickeltes Typensatzprogramm
- Ebenfalls rechteck-orientiert
- Tipp: Nicht mit LaTeX verwechseln!
(LaTeX entspricht eher DocBook, TeX entspricht XSL-FO)

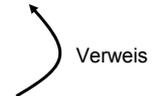
XSL-FO: Ein Minimalbeispiel Die FO-Dokumentenstruktur

- "Hello World"

```
<?xml version="1.0" encoding="iso-8859-1"?>
<fo:root xmlns:fo="http://www.w3.org/1999/XSL/Format">

  <fo:layout-master-set>
    <fo:simple-page-master master-name="my-page">
      <fo:region-body margin="1in"/>
    </fo:simple-page-master>
  </fo:layout-master-set>

  <fo:page-sequence master-reference="my-page">
    <fo:flow flow-name="xsl-region-body">
      <fo:block>Hello, world!</fo:block>
    </fo:flow>
  </fo:page-sequence>
</fo:root>
```



Verweis

- Mini-Demo
 - Anzeige der resultierenden PDF-Datei (tut20.pdf)
- Erkenntnisse und Fragen aus dem Beispiel
 - **root**
 - Ein reiner Container, nützlich i.w. für Namensraumdeklarationen
 - Genereller Dokumentenaufbau?
 - **layout-master-set**
 - Zur Seitengestaltung, immer erforderlich
 - Alternativen zu "simple-page-master"?
 - Definition des "region"-Begriffs?
 - **page-sequence**
 - Was bedeutet "flow"? Was ist mit "xsl-region-body" gemeint?
 - Gestaltungsmöglichkeiten des Textes?

- Genereller Dokumentenaufbau in DTD-Notation

```
<!DOCTYPE fo:root ... [
  <!ELEMENT fo:root (
    fo:layout-master-set,
    fo:declarations?,
    fo:page-sequence+) >

  <!ELEMENT fo:declarations (
    fo:color-profile)+ >

  <!ELEMENT fo:page-sequence (
    fo:title?,
    fo:static-content?,
    fo:flow)>

  <!ELEMENT fo:layout-master-set(
    fo:simple-page-master |
    fo:page-sequence-master )+ >>

  <!-- mind. ein fo:simple-page-master -->
```

- Genereller Dokumentenaufbau in DTD-Notation (Forts.)

```
<!ELEMENT fo:simple-page-master (
    fo:region-body,
    fo:region-before?, fo:region-after?,
    fo:region-start?, fo:region-end?) >

<!ELEMENT fo:page-sequence-master (
    fo:single-page-master-reference |
    fo:repeatable-page-master-reference |
    fo:repeatable-page-master-alternatives)+>

<!ELEMENT fo:repeatable-page-master-alternatives (
    fo:repeatable-page-master-reference)+ >

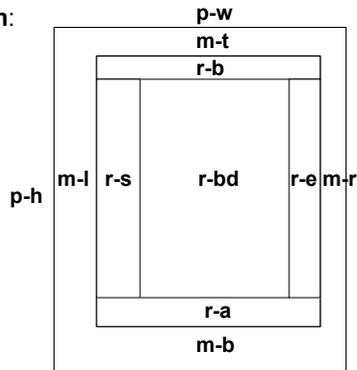
]>
```

**XSF-FO: Das Seitenmodell.
Page masters und Regionen**

Das FO-Seitenmodell und seine Regionen

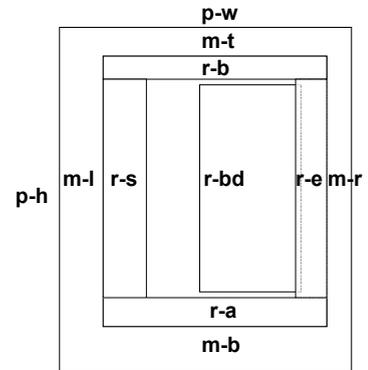
- Eine Seite besitzt eine **Höhe** und **Breite** und besteht aus **4 Rändern** und **5 Regionen**:

- page-height, page-width
- reference-orientation
- margin-top, margin-bottom, margin-left, margin-right;
- margin (Kurzform)
- region-before, region-after, region-start, region-end, region-body (Darstellung hier für westliche Sprachen: lr-tb)



Hinweis:

- Die Randregionen konkurrieren um denselben Platz wie region-body!
- Ob Lücken oder Überdeckungen entstehen, bestimmt der Anwender durch die Ränder von region-body.



Einfaches Pagemaster-Beispiel

```
<fo:layout-master-set>
  <fo:simple-page-master master-name="DIN-A4-P"
    page-height="29.7cm" page-width="21cm"
    margin-top="1.5cm" margin-bottom="2cm"
    margin-left="2.5cm" margin-right="2cm">
    <fo:region-body/>
    <!-- Andere "regions" sind optional - hier ausgelassen. -->
  </fo:simple-page-master>
</fo:layout-master-set>
```

- DIN A4 Hochformat („portrait“)
- Extra breiter Heftrand links
- region-body füllt den gesamten Platz innerhalb der Ränder aus.

Testfragen

- Wo liegt welche Region im Fall
a) writing-mode="rl-tb"
b) writing-mode="tb-rl"
c) writing-mode="lr-tb" mit reference-orientation="90" ?

An der Tafel diskutieren!

- Bem. zu den o.g. Werten von "writing-mode": (Skizze!)
- Erst "inline-progression-direction",
- dann "block-progression-direction"

Regionen

- Normalerweise erfolgt die Ausgabe in "region-body".
- Die vier anderen Regionen lassen sich aber separat davon ebenfalls mit Inhalten füllen!
- Typische Objekte in den Randregionen:
 • Rechts & Links:
 - Randnotizen, Änderungsvermerke und -balken
 • Oben & unten:
 - Kapitelüberschriften, Seitennummern, Copyright-Notizen, Firmenlogo, ...
- Achtung:
 • Regionen überlappen miteinander, denn Lage und Größe von region-body sind unabhängig von den 4 anderen regions!
 - Dies lässt sich nutzen z.B. für overlay-Effekte ...
 - ... oder verhindern durch geeignet große Ränder in region-body!

Pagemaster-Beispiel mit Raum für header+footer

```
<fo:layout-master-set>
  <fo:simple-page-master master-name="DIN-A4-02"
    page-height="29.7cm" page-width="21cm"
    margin-top="1cm" margin-bottom="1cm"
    margin-left="2.5cm" margin-right="2cm">
    <fo:region-body margin-top="2.5cm" margin-bottom="2cm" />
    <fo:region-before extent="2cm" />
    <fo:region-after extent="1.5cm" />
  </fo:simple-page-master>
</fo:layout-master-set>
```

TEST:
Wie groß ist region-body?

- Oben und unten steht nun Raum für header & footer zur Verfügung
- region-body lässt zusätzlich noch 0.5 cm Platz nach oben und unten.

Möglichkeiten mit *Page Masters*

– **Abfolge** von Seiten mit verschiedenen *Layouts*, etwa

- Deckblatt
- Danksagung
- Inhaltsverzeichnis: 1. Seite und Folgeseiten
- Normales Kapitel: 1. Seite, Folgeseiten
- Glossar: 1. Seite, Folgeseiten
- Index: dito (z.B. zweispaltig)

– **Bedingte Formatierung**

- `odd-or-even` mit Werten `odd - even - any`
- `page-position` mit Werten `first - last - rest - any`
- `blank-or-not-blank` mit Werten `blank - not-blank - any`

XSF-FO: Umgang mit Blöcken und Gebieten

Eigenschaften von Blöcken und Gebieten (*areas*)

– **Hintergrund**

- `background-color` (red, #FF008E, transparent)
- `background-image` (URI, none)
- `background-attachment` (fixed oder scroll)
- `background-position` (center, left, right, middle, top, bottom, a coordinate)
- `background-repeat` (repeat, no-repeat, repeat-x, repeat-y)
- Beispiel:

```
<fo:block background-image="/bg/paper.gif"
          background-position="0,0"
          background-repeat="repeat"
          background-color="white">
  Irgend ein Text...
</fo:block>
```

Eigenschaften von Blöcken und Gebieten (*areas*)

– **Ränder**

- Zwei (!) Sätze von Attributen:
- 1. CSS-kompatibler Stil:
`margin-top, margin-bottom, margin-left, margin-right; margin`
- 2. FO-Stil, richtungsneutral:
`space-before, space-after; start-indent, end-indent`

Eigenschaften von Blöcken und Gebieten (*areas*)

– Rahmen: Insgesamt 31 Eigenschaften!

- Sei $X \in \{\text{before, after, start, end, top, bottom, left, right}\}$
- Redundanz wegen CSS / FO-Stil beachten!
- Farbe: border-color, border-X-color. Default: black
- Breite: border-width, border-X-width. Default: medium
- Stil: border-style, border-X-style. Default: none
- Kurzformen: border, border-color, border-width, border-style, border-top, border-bottom, border-left, border-right

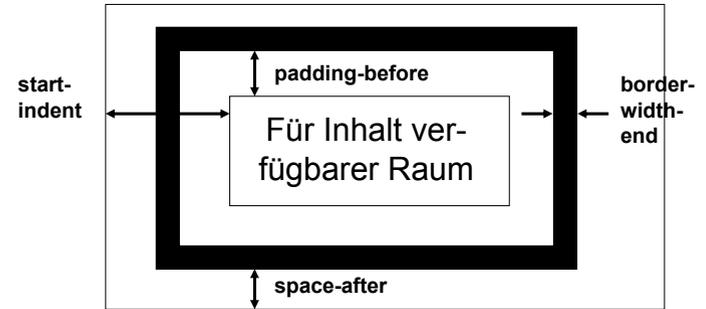
– Beispiel: Ein blauer Rahmen mit 2 Pixeln Breite

```
<fo:block
  border-before-color="blue" border-before-width="2px"
  border-after-color="blue" border-after-width="2px"
  border-start-color="blue" border-start-width="2px"
  border-end-color="blue" border-end-width="2px">
  Irgend ein Text.
</fo:block>
```

Eigenschaften von Blöcken und Gebieten (*areas*)

– Abstände (*padding*)

- 8 Attribute: padding-X, mit X wie oben definiert
- Modell dazu, Fall lr-tb (Rahmen übertrieben):



Eigenschaften von Blöcken und Gebieten (*areas*)

– Inhalt: Größenangaben

- height, width
- max-height, max-width
- min-height, min-width
- Der Wert "auto" sorgt für automatische Größenanpassung des Blocks an seinen Inhalt. max-* bzw. min-* können obere bzw. untere Grenzen für diese Automatik setzen.

– Inhalt: Overflow-Verhalten

- Anzeigen, Abbruch, Clipping (wo?), etc. sind steuerbar

– Inhalt: reference-orientation, writing-mode

- Diese bereits angetroffenen, normalerweise globalen Eigenschaften können lokal überschrieben werden.

Beispiele zur Gestaltung von Blöcken:

– Tutorial-46.fo / pdf

- Quelltext
- Wirkung in PDF-Ausgabe per Acrobat Reader

– Man beachte, dass FOP manche FO-Elemente bzw. Attribute nicht oder nur unvollkommen unterstützt.

- Dies ist freilich abhängig von der FOP-Version (Demo!)

XSL-FO: Inline- und Textgestaltung

- Eigenschaften von *Inline*-Elementen
 - Ziel: Gestaltung von Texten
 - Methode:
 - Verwaltung von Eigenschaften mittels Attributen
 - Vererbungskonzept
 - Bem.: Es gibt mehr als 200 solche Eigenschaften!
Wir werden sie hier auch nicht ansatzweise besprechen...
 - Unterscheide Attribute auf Block- bzw. Paragraphen-, Satz-, Zeilen-, Wort- und Buchstaben-Ebene. Beispiele:
 - Block-Ebene: Satzumbbruch-Steuerung, Einrückungen
 - Satz-Ebene: Zeichenabstände, Wortabstände
 - Zeilen-Ebene: Zeilenabstände, Textausrichtung
 - Wort-Ebene: Steuerung zur Wörtertrennung
 - Zeichen-Ebene: Font, Farbe, verschiedene Text-Attribute

- Das Element **fo:inline**
 - Reines *Container*-Element
 - Dient der Aufnahme zahlreicher Attribute
 - Konstrukt zur Umsetzung des Vererbungskonzepts

- Beispiel:

```
<fo:block font-family="Times" font-size="14pt"
  font-style="italic">
  <fo:inline color="red">H</fo:inline>ello,
  <fo:inline font-weight="bold">world!</fo:inline>
</fo:block>
```

- Beispiele zur Gestaltung von *Inline*-Elementen, insb. von Texten:
 - Tutorial-33.fo / pdf
 - Mehr Quelltexte
 - Wirkung in PDF-Ausgabe per Acrobat Reader
 - Man beachte, dass FOP manche FO-Elemente bzw. Attribute nicht oder nur unvollkommen unterstützt.

XSF-FO: Grafiken

XSL-FO

- Einbettung fremder Grafiken (Bitmaps)
 - Vergleichsweise einfach - bitte selbst nachlesen!
 - Element fo:external-graphics
- XSL-FO und SVG (**Demo: fo+svg01.fo**)

```
<fo:block>
  Straßenschild "Einfahrt verboten":
  <fo:instream-foreign-object>
    <svg:svg xmlns:svg="http://www.w3.org/2000/svg"
      height="100" width="100" viewBox="-50 -50 100 100">
      <svg:circle r="50" style="fill:red; stroke:none"/>
      <svg:rect x="-40" y="-10" width="80" height="20"
        style="fill:white; stroke:none"/>
    </svg:svg>
  </fo:instream-foreign-object>
</fo:block>
```

XSF-FO: Sonstiges

XSL-FO

- Hinweise auf weitere FO-Eigenschaften und Möglichkeiten

– Tabellen

- Das Tabellenmodell von FO ist dem von CSS sehr ähnlich:

```
<!ELEMENT fo:table-and-caption (fo:table-caption?, fo:table)>
<!ELEMENT fo:table (fo:table-column*, fo:table-header?,
  fo:table-footer?, fo:table-body+)>
<!ELEMENT fo:table-body (fo:table-row+ | fo:table-cell+)>
```
- Eine Zelle schließlich enthält einen Block.
- **Projekt: Für Ergebnislisten empfohlen!**

– Listen

- fo:list-block, bestehend aus fo:list-item-Elementen, darin:
- fo:list-item-label, fo:list-item-body

- Hinweise auf weitere FO-Eigenschaften und Möglichkeiten
 - **Fußnoten**
 - **"Floats"**
 - *out-of-line* Objekte wie Tabellen, Abbildungen, separate Textkästen, die möglichst auf die Seite ihrer Entstehung gelangen sollen, deren genaue Lage der Prozessor aber bestimmen kann.
 - **Links**
 - **"Leaders"** (z.B. für Inhaltsverzeichnisse: Kap. 5 30)
 - **Trennlinien ("rules")**
 - Steuerung auditorischer Ausgabe
 - Möglichkeiten wie bei CSS

- Hinweis auf Testmaterial
<http://www.renderx.com/featurestest.html>
 - Hier finden Sie zahlreiche XML-Quelldateien, passende XSL-Transformationsdateien, resultierende FO-Dateien und deren Darstellung als PDF zu praktisch allen Möglichkeiten von XSL-FO
- XSL-FO: Quo vadis?
 - Ziele des 1.1-Releases:
 - Neue Funktionalität bei Änderungsmarkern, Indexangaben, mehreren „flows“ und Lesezeichen.
 - Erweiterte Funktionalität bei der Skalierung von Grafiken, bei Markern, insb. im Zusammenspiel mit Tabellen.
 - Berücksichtigung inzwischen bekannt gewordener Korrekturen.

**XSF-FO: Hinweise
zum Praktikumseinsatz**

- FO-Prozessor
 - Zur Verfügung steht der (in Java implementierte) Open-Source Prozessor FOP V. 0.20.5, vgl. DocBook-Übung
 - Alternativ: Aktuelle Version, V 0.92.beta (05/2006)
 - Einschränkungen:
 - Zahlreiche Eigenschaften von XSL-FO hat FOP noch nicht implementiert und meldet dies.
 - Andere Eigenschaften hat FOP ebenfalls nicht (oder fehlerhaft) implementiert, meldet dies aber nicht!
 - **Nützlich:** Beispiele aus Quelle [2] funktionieren mit FOP 0.20.4 oder enthalten Hinweise auf FOP-bedingte Grenzen

- **Selbst-Studium**
 - Dieses Vorlesungskapitel ist nur ein Einstieg in XSL-FO.
 - Konzepte und Grundbegriffe stehen hier im Vordergrund
 - Nutzen Sie die Anwesenheitszeit zur Klärung von Verständnisfragen, denn:
 - **Es wird erwartet, dass Sie sich die Einzelheiten, die zur Bearbeitung der Praktikumsaufgaben erforderlich sind, selbst aneignen.**
 - Zur Verfügung stehen Ihnen die angegebenen Quellen.
 - Besonders hilfreich: [2] und [3]
 - Hinweis: Auch die W3C-Spezifikation [1] ist durchaus lesbar. Sie enthält zahlreiche nützliche Beispiele.
 - Ferner hilfreich:
 - Analyse des FO-Outputs der DocBook-Übung

- **Grafiken**
 - Urkunden
 - Verwenden Sie das FH-Logo (SVG-Grafik) für ein Hintergrundbild
 - Das Hintergrundbild soll nur dezent erscheinen (eine Art Wasserzeicheneffekt, vielleicht mit einen sehr hellen Grauton erreichbar)
 - Es soll andererseits groß ausfallen und nur einmal erscheinen
 - Ergebnislisten
 - FH-Logo in kleiner Form in den Seitenkopf?
 - Quelle
 - Verwenden Sie als Quelle die SVG-Datei aus dem Praktikumsversuch zu SVG des XML-Kurses im WS05/06
 - Wird auf Anfrage auch gestellt.
 - **Achtung: Grafik in FO-Ausgaben = Prio-3 Projektteil.**