

The Social Life of Forms

How to speak like you're from Boston (Bahstin)

1. Loss of post-vocalic 'r'

1. Park the car in Harvard Yard → Pahk the ca in Havad Yad

2. But 'r' is preserved if followed by vowel

1. Is the bar open? → iz de bar opin? Or iz de ba ropin?

3. But 'r' is added to the end of some words, for no particular reason:

1. Idea → ideer

2. Korea → Koreer

3. But note that Career → Korea, per rule 1.

Obligatory Quote from someone smarter than me

“XML lets organizations benefit from structured, predictable documents. Thus, XML breeds forms. QED.”

-David Weinberger

Obligatory Block Diagram

1 H																	2 He
3 Li	4 Be											5 B	6 C	7 N	8 O	9 F	10 Ne
11 Na	12 Mg											13 Al	14 Si	15 P	16 S	17 Cl	18 Ar
19 K	20 Ca	21 Sc	22 Ti	23 V	24 Cr	25 Mn	26 Fe	27 Co	28 Ni	29 Cu	30 Zn	31 Ga	32 Ge	33 As	34 Se	35 Br	36 Kr
37 Rb	38 Sr	39 Y	40 Zr	41 Nb	42 Mo	43 Tc	44 Ru	45 Rh	46 Pd	47 Ag	48 Cd	49 In	50 Sn	51 Sb	52 Te	53 I	54 Xe
55 Cs	56 Ba	*	72 Hf	73 Ta	74 W	75 Re	76 Os	77 Ir	78 Pt	79 Au	80 Hg	81 Tl	82 Pb	83 Bi	84 Po	85 At	86 Rn
87 Fr	88 Ra	**	104 Rf	105 Db	106 Sg	107 Bh	108 Hs	109 Mt	110 Ds	111 Rg	112 Uub	113 Uut	114 Uuq	115 Uup	116 Uuh	117 Uus	118 Uuo
* Lanthanides		57 La	58 Ce	59 Pr	60 Nd	61 Pm	62 Sm	63 Eu	64 Gd	65 Tb	66 Dy	67 Ho	68 Er	69 Tm	70 Yb	71 Lu	
** Actinides		89 Ac	90 Th	91 Pa	92 U	93 Np	94 Pu	95 Am	96 Cm	97 Bk	98 Cf	99 Es	100 Fm	101 Md	102 No	103 Lr	

Obligatory reference to antiquity

Aramaic tax receipt from 355BC

On the 16th of Tammuz,
year 4 of King Artaxerxes,

Halfat brought

barley: 1 kor, 12 seah, 3 qab

wheat: 1 kor, 5 seah, 4 qab

15 day returns with receipt. Opened
packages will receive store credit
only. Not returns on display models
or other as-is items.



Times have not changed much

- Then as now, the greatest producers and consumers of documents are bureaucracies:
 - Government at all levels (federal, state, municipal)
 - Insurance
 - Healthcare
 - Banking
 - And business that deal with the above:
 - Regulated industries (securities, etc.)
 - Corporate HR departments

In the beginning...

- First attested written use of the word “form” is in a critical portrait of industrial era bureaucracy – Charles Dickens's *Little Dorrit* (1855)
 - The waiting-rooms of that Department soon began to be familiar with his presence, and he was generally ushered into them by its janitors much as a pickpocket might be shown into a police-office; the principal difference being that the object of the latter class of public business is to keep the pickpocket, while the Circumlocution object was to get rid of Clennam. However, he was resolved to stick to the Great Department; **and so the work of form-filling, corresponding, minuting, memorandum-making, signing, counter-signing, counter-counter-signing, referring backwards and forwards, and referring sideways, crosswise, and zig-zag, recommenced.**

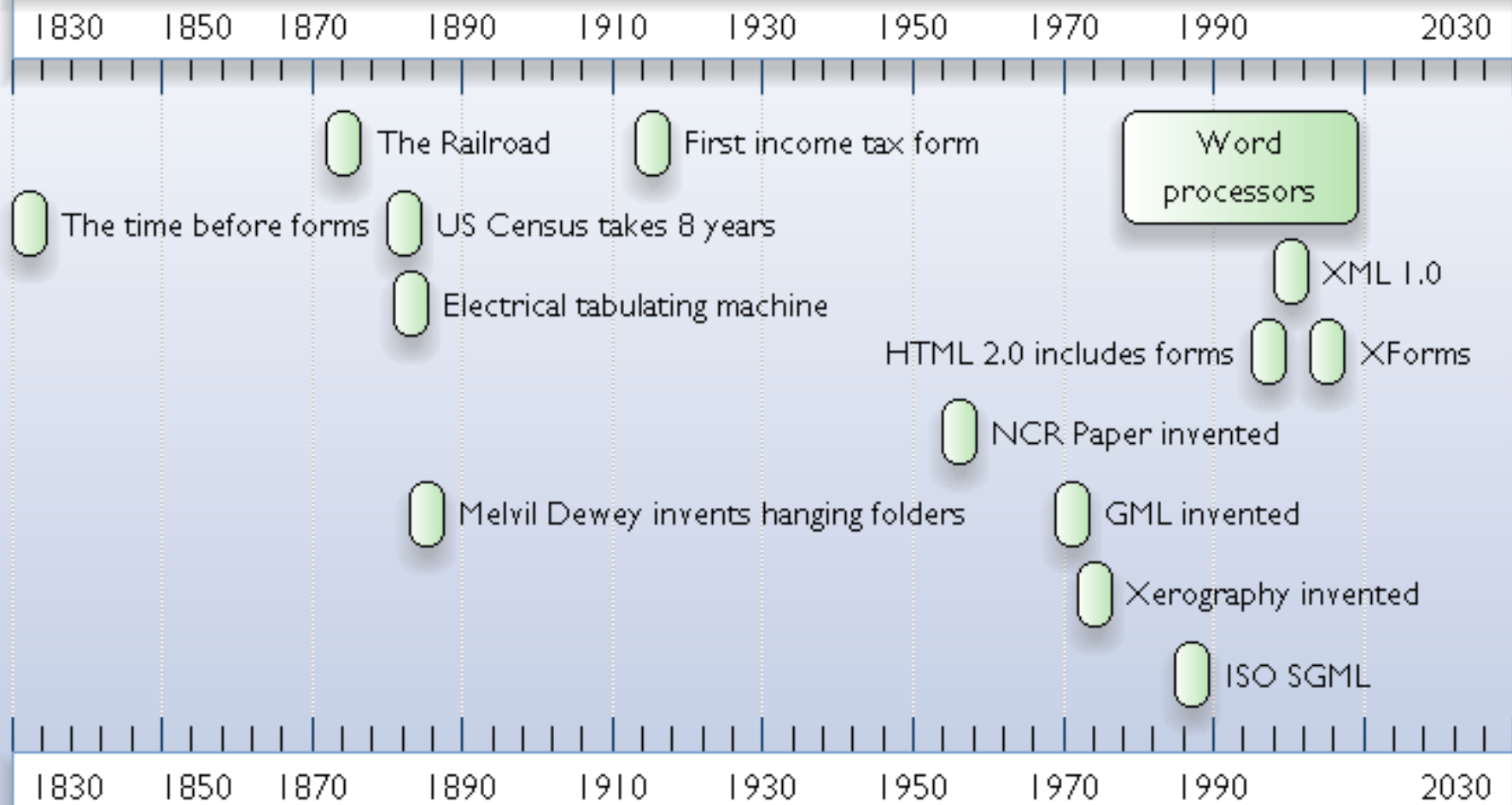
153 years ago

Historic forms cost drivers

- **Production**
- **Distribution**
- **Data entry**
- **Submission**
- **Reproduction**
- **Retrieval**
- **Processing**

We can see how these have been addressed over time...

History of Forms



Where do we stand today with electronic forms?

- **Production** of forms today requires a team of highly-skilled specialists rather than a entry level office clerk – so form production costs are up. Not just a technical issue:
 - Requirements, perhaps internally developed, or statutory
 - Data model definition, perhaps to match corporate or industry schemas
 - Layout of form elements
 - Coding of application logic
 - Reviews: legal, department, accessibility, usability, test
 - Change management / archiving
- **Distribution** of forms is practically free with the Internet
- **Filling out** of forms hasn't changed appreciably in 100 years. We type rather than write (sometimes).
- **Submission** is also nearly free on the internet, but often data is submitted on paper and then entered into computer
- **Reproduction** can be had for the price of storage
- **Retrieval** is much cheaper, no more shelves of forms or tapes
- **Processing**, where it can be automated is cheaper, but still many business processes are not automated or automated in unconnected applications

New challenges

- Scalability – US government has 17,000+ online forms today. How does one manage design, approval and deployment on this scale, especially with a high rate of change, both anticipated (annual) or on-demand (new legislation, court decision or executive order)
- Collaboration: forms development is a cross-team development activity, very much like software development. How do the tools support this?
- Multi-device, multi-channel, multi-mode delivery – paper isn't enough. HTML isn't enough. We want it at our fingertips via mobile devices, via IM, via email, phone, etc.
- Search -- What is easier, having the user find the form, or having the form find the user?
- Unique challenges of G2C communications – can't assume tech savvy users with latest browser upgrades and high end systems with broadband connections. Also, accessibility requirements. In a census or on-line voting, lack of access is not just a technical question, it is a civil rights issue.

Typical Electronic Forms Features

- Fields

- have data types

- have captions/labels

- have constraints

- have error messages

- can be grouped

- have a control type (listbox versus group control versus edit control, etc.)

- can be active/inactive (grayed out)

- can be read-only or editable

- Forms

- contain one or more fields

- also contain static text

- dependencies between fields (calculation chains)

- data awareness (receive data from a database)

- submission – to database, to web server, to web service, via email

- schema awareness

Electronic Forms – What's out there?

- Proprietary solutions
 - InfoPath
 - XFA
- Open solutions
 - Form document exchange via email
 - HTML
 - XForms

XForms Implementations

Open Source

Firefox extension
Convex
formsPlayer

OpenOffice
X-Smiles
DENG

Browser-based

Standalone

Lotus Forms
FormFaces

Lotus Forms
picoForms

Proprietary

Forms in HTML

```
<form name="myForm" method="get" action="/doIt">
```

```
<input type="text" name="fname">
```

.

.

.

```
<input type="submit" name="button">
```

```
</form>
```

Problems with HTML Forms

- Poor integration with XML – you essentially can only submit name/value pairs
- Default values are hard-coded
- Poor separation of data, UI and behavior

- Result is that “real-world” forms require extensive scripting (for field validations, calculations, error messages, etc).

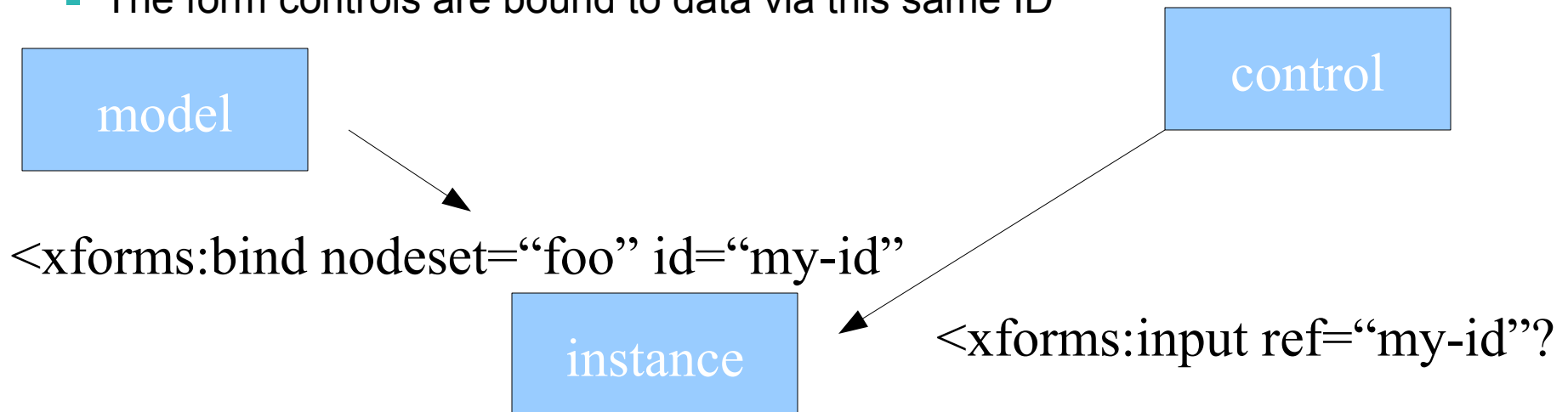
- Complexity → Cost

Enter XForms

- Created in W3C (2000-2003)
- A break from HTML forms
- A number of best practices included
 - XML is supported directly, not as an afterthought
 - XML Schema-aware
 - Model-View-Control separation – the killer design pattern for flexible application development
 - Uses “the spreadsheet algorithm”
 - Designed for accessibility from the start
 - Declarative business rules
 - Scripting not required “most of the time”

The parts of an XForm

- `<xforms:model>` -- model in the model-view-controller sense == the runtime state of the form
 - `<xforms:instance>` -- an inline or linked XML document instance, typically used as the default values for the form
 - `<xforms:bind>` -- sets type, constraints, read-only state, etc., (the “Model Item Properties”)
 - `<xforms:submission>` -- submits the form, via HTTP GET/POST/SOAP
-
- The model binds data from the instance to an ID
 - The form controls are bound to data via this same ID



Let's do a demo!