

# **Device-Independence with UIML**

## **(User Interface Markup Language)**

*Marc Abrams*  
[marc@harmonia.com](mailto:marc@harmonia.com)

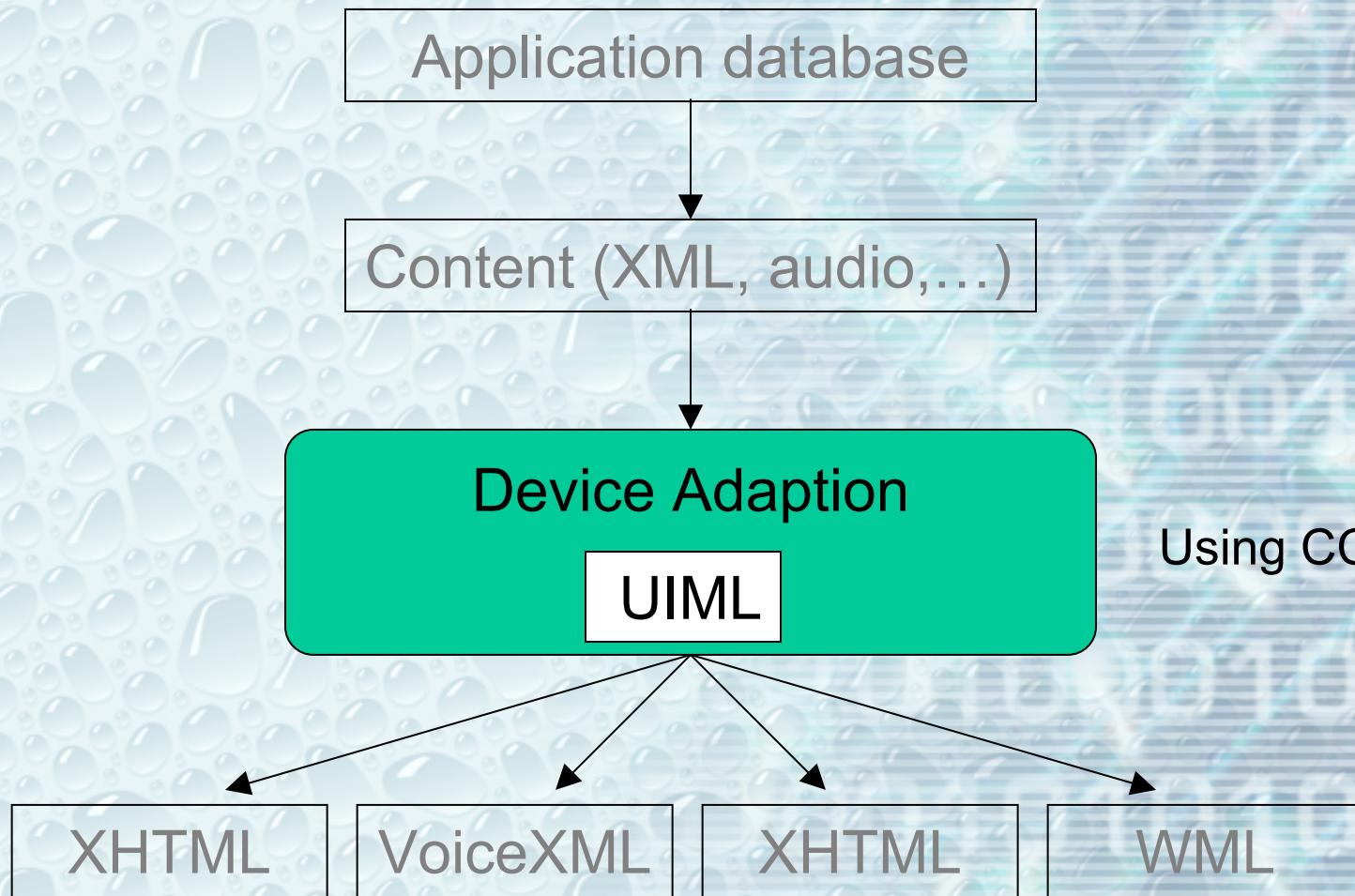
*uiml.org*

*W3C Workshop on Web Device Independent Authoring  
October 2000*

# Quotes from Position Papers

- New XML language should
  - Allow author-once-deploy many scenarios
  - Achieve clean separation between behavior, content, presentation [Ruud Siebelink]
- Issue is *interaction* not *presentation* [Paul Smethers, WAP]
- Ideal solution: write well-formed code once [Jansen]
- Semantics [meaning of Web content] must be made clear at primary design level [William Loughborough]
- Must adapt to new devices not envisioned [Ralph Case, Stephane Maes]

# Where UIML Fits In\*



\* Diagram from Dave Raggett's talk

# UIML... One Part of a Solution

- One canonical representation of UI for any device, language, OS, UI-metaphor
- 3+ years in development at Harmonia, Center for HCI at Virginia Tech
- Tools downloaded in 40+ countries
- Can be compiled to lots of things
- Anyone can freely implement UIML
- Objective is open standard

# Problem with Existing Approaches

- Suggested way to annotate existing markup:

~~<card>~~

<select class="DISPLAY\_SMALL">

...

~~</card>~~

# Key Concept: UIML is a “Meta” Language

- XML
  - Doesn't define tags (<P>,...)
  - Must add doc type definition to make it useful
  - No need to change XML as new tag sets invented
- UIML
  - Doesn't define tool-kit specific tags (<Menu>,...)
  - Uses a few powerful tags (<part>, <property>,...)
  - Must add toolkit vocabulary to make it useful
  - No need to change UIML as new devices invented

# UIML Model

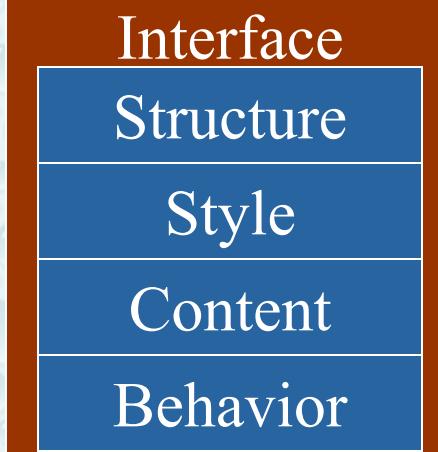
Underlying principle of single authoring is MVC [to separate content/structure] [Ralph Case, Stephane Maes]

Device / Platform  
UI Metaphor

Application

Presentation

Logic



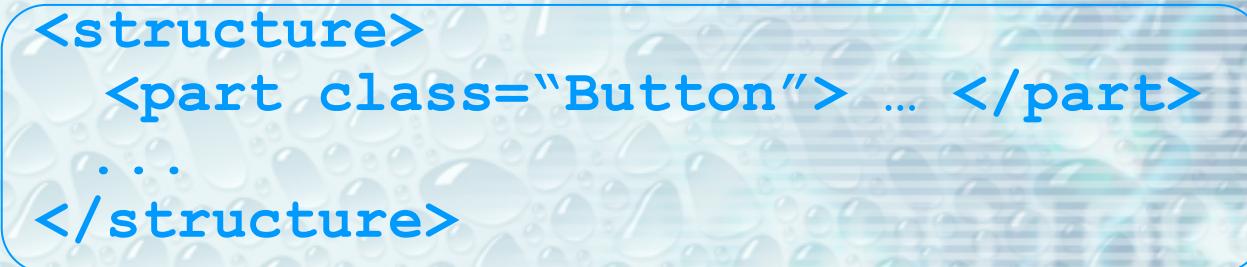
6-way separation of UI description (vs. 3-way MVC)

# UIML Skeleton – Part 1

```
<?xml version="1.0" ... ?>
<uiml version="2.0">

<interface>
  <structure>
    <part class="Button"> ... </part>
    ...
  </structure>
</interface>

</uiml>
```



*What parts comprise the UI &  
what's their relationship?*

# UIML Skeleton – Part 2

```
<?xml version="1.0" ... ?>
<uiml version="2.0">

  <interface>
    <structure>...</structure>
    <stylestyle
```

*What presentation style  
for each part?*



# UIML Skeleton – Part 3

```
<?xml version="1.0" ... ?>
<uiml version="2.0">

  <interface>
    <structure>...</structure>
    <style>...</style>
    <content>...</content>
  </interface>

</uiml>
```

*What content for each part?  
(text, sounds, image, ...)*



# UIML Skeleton – Part 4

```
<?xml version="1.0" ... ?>
<uiml version="2.0">

  <interface>
    <structure>...</structure>
    <style>...</style>
    <content>...</content>
    <behavior>...</behavior>
  </interface>

</uiml>
```

*What behavior do parts have?*

# UIML Skeleton – Part 5

```
<?xml version="1.0" ... ?>
<uiml version="2.0">

  <interface>
    <structure>...</structure>
    <style>...</style>
    <content>...</content>
    <behavior>...</behavior>
  </interface>
  <peers>...</peers>
</uiml>
```

How to connect to outside world?  
(business logic, UI toolkit object)

# <peers> Maps Classes to Targets

```
<d-class name="JButton" ... maps-to="javax.swing.JButton">  
...  
</d-class>
```

*Versus*

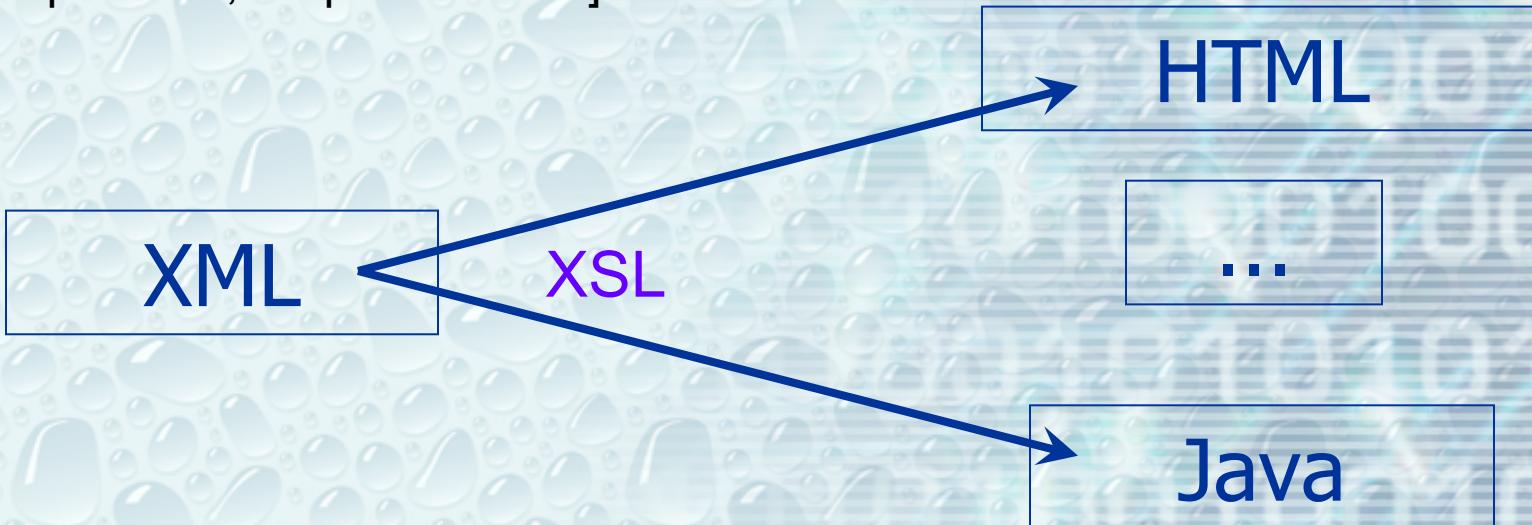
```
<d-class name="JButton" ... maps-to="html:input">  
...  
</d-class>
```

*This part is written once, like a device driver for an OS.  
Events and calls to outside world handled similarly.*

# NxM Problem (Old Way)

- App composed on
  - $M$  “pages”
  - accessed via  $N$  devices
  - requires  $N \times M$  authoring steps

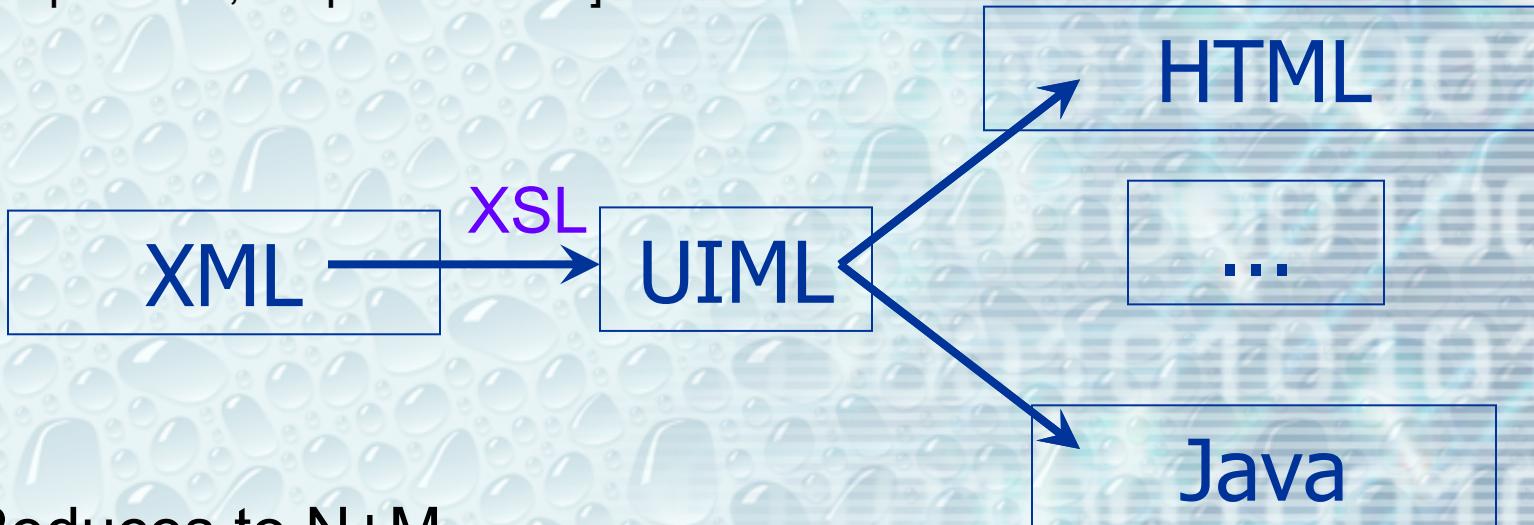
[Ralph Case, Stephane Maes]



# NxM Problem (New Way)

- App composed on
  - $M$  “pages”
  - accessed via  $N$  devices
  - requires  $N \times M$  authoring steps

[Ralph Case, Stephane Maes]



Reduces to  $N+M$

# UIML Permits Development Continuum

Richest user experience

Fastest to build

↑  
Use  
device-specific  
vocabulary:

`<part Class="JPopupMenu">`  
*versus*  
`<part Class="Select">`

↑  
Use  
generic  
vocabulary:

`<part Class="Menu" >`

# UIML Permits Families of UIs

**Harmonia Finance Manager**

Account No.	11111	From	12/13/97	To	01/12/98	Change Statement
Date	Check No./Trans...	Payee/Location	Withdraw	Deposit	Balance	
12/6/97	205	Giant Food	\$15.50	-	\$605.87	
12/7/97	206	Media General C...	\$32.55	-	\$573.32	
12/8/97	207	Washington Gas ...	\$103.00	-	\$470.32	
12/15/97	2100	Circuit City	\$225.75	-	\$244.57	
12/15/97	ATM Withdrawl	Fair Oaks Mall	\$75.00	-	\$169.57	
12/17/97	DEPOSIT	Branch 1103	-	\$984.54	\$1,154.11	
12/20/97	210	Chilbank VISA	\$260.07	-	\$893.14	
12/23/97	2108	Giant Food	\$100.57	-	\$792.57	
1/0/98	ATM Withdrawl	3022 Sudley Rd.	\$125.00	-	\$677.57	

Current Balance: \$677.57

ATM

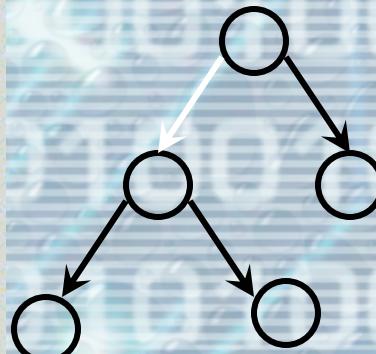
Date: January 14, 1998

Stephen M. Williams  
10 Williamson Road  
Roanoke, VA 24027

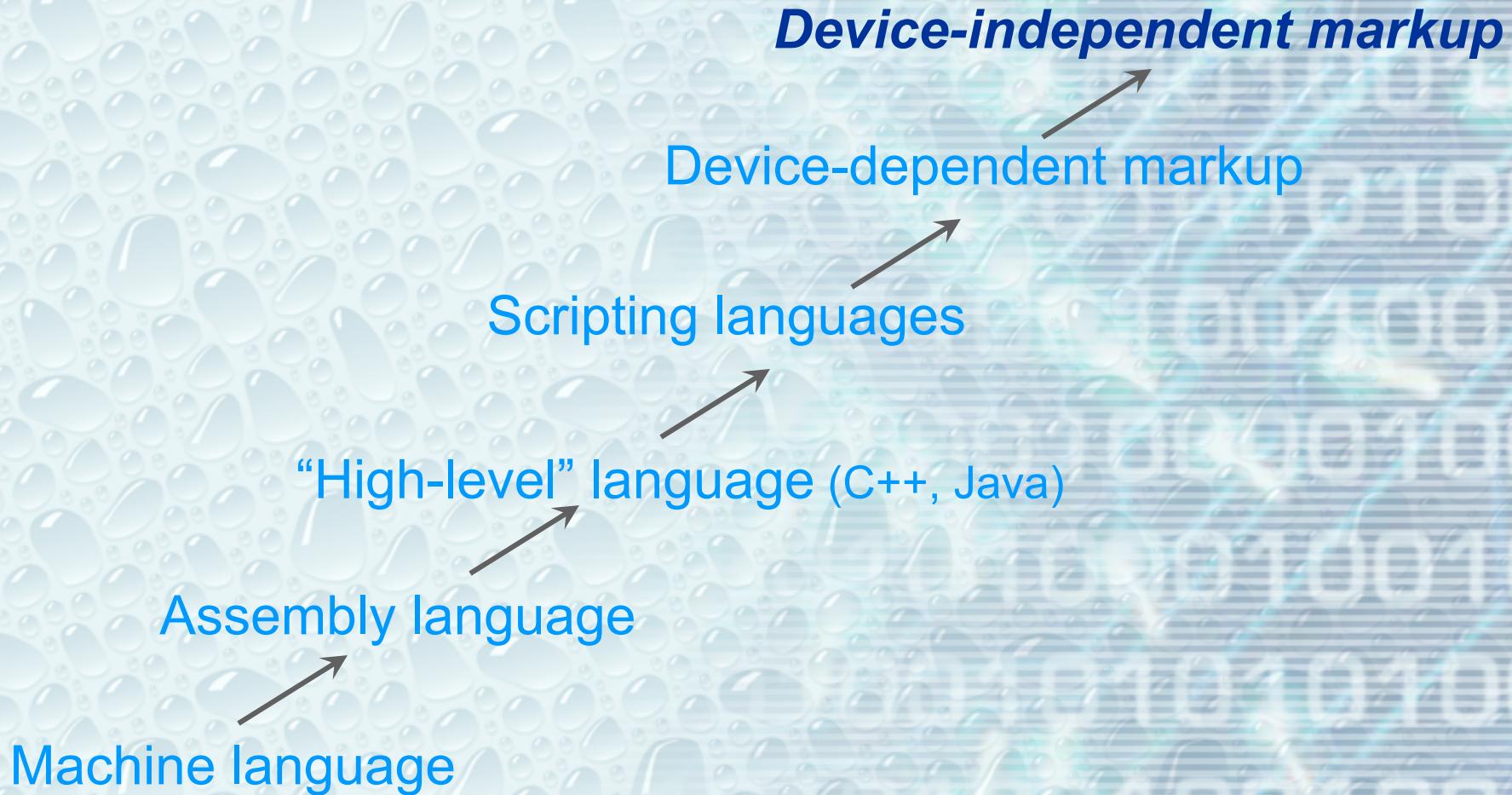
Pay To: Washington Gas \$ 142.44

Amount: One hundred forty two and 44/100 Dollars

Memo:  Sign and Send



# Another Perspective...



# Still... UIML is Not a Silver Bullet...

Some open problems:

- Aid/enforce accessibility guidelines [Jon Wu]
- Support auto adaptation/personalization  
[Ruud Siebelink]
- Reorganizing UI:
  - Many apps will need to be re-designed entirely  
[Guido Grassel]
  - 1 page in a desktop Web browser might be split  
into 2 screens for TV [Peter Ferne]

# For More Info

Visit [uiml.org](http://uiml.org)

Upcoming: European Workshop on UIML –  
January 2001 in Paris