Assistive Technologies for STEM Subjects From Bitmap Graphics to Fully Accessible Chemical Diagrams

Volker Sorge

Scientific Document Analysis Group
School of Computer Science
University of Birmingham

Progressive Accessibility Solutions
Birmingham, UK
progressiveaccess.com

3rd International Workshop on Digitization and E-Inclusion In Mathematics and Science, 2016, Shonan, 4 February 2016

The Future of STEM Content

- Like for most content the future is the Web
  - Ubiquitous
  - Flexible
  - Adaptable
  - Independent
  - And: quick and easy to publish
- Two issues (also for accessibility):
  - Lack of web support for STEM
  - Proliferation of short lived content

Problems with STEM Content on the Web

- Full of specialist notations, formulas, diagrams, charts, etc.
- Authors do not understand the web: Content is geared towards print.
- This makes it already difficult to work with in standard use cases let alone in the accessibility use case.

- So what do we need?
- Specialist web standards for all types of STEM content:
  - Chemistry, Biology, Computer Science, ...
  - From STEM to STEAM: Music, Manuscripts, ...

NO! We do not need specialist web standards!

A Warning Example: The Failure of MathML on the Web

- MathML is officially part of the HTML5 standard
- Mathematics should be formatted in (presentation) MathML.
- Generally this is not the case: Instead it is given as LaTeX or ASCIIMath.
- MathML has very limited support from Browser vendors
  - Two incomplete implementations: FireFox (Gecko), Safari (WebKit)
- MathML spec is seriously outdated
  - Refuses to take modern web technology into account!

The Role of Polyfill Solutions

- With technologies like SVG, HTML5/Canvas, CSS/Houdini, there are (nearly) unlimited possibilities to shape web content
- There is no need and no desire for specialist standards
- Specialist content will be treated more and more by bespoke, but universal rendering solutions: Polyfills
- MathJaX is an example that has filled the need for Mathematics rendering
- Others include graphics libraries like: JSxGraph, D3js,. . .

What about Accessibility?

- Semantics has to be provided regardless of the underlying implementation
- WCAG 2.0 is not sufficient: work on more ARIA, standard APIs, etc.
- Polyfills start using Universal design principles
- This can solve the problem of making ephemeral material accessible

Accessibility of STEM Material

- In the light of a fleeting medium like the Web traditional techniques fail
  - Audio recordings, tactile graphics, German film, Physical Models, Specialist translation service
- Lack of timeliness
- Often lack of resources in the real world
- But web is ideal to further learner independence

Case Study on (Chemical) Diagrams

- Diagrams are very important for teaching STEM subjects
  - Geometry, Physics, Chemistry, Biology, ...
- Chemical diagrams (depictions of molecules) are ubiquitous in teaching material on chemistry, biosciences, life sciences.
  - GCSE and A-levels teaching
  - undergrad curriculum
  - research publications

Accessibility of Digital Diagrams

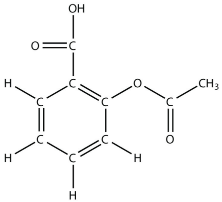
- Bitmaps are simply inaccessible and ALT texts are generally not enough
- Overlay bitmaps for tactile or touch exploration
- SVG with screen reading software
  - Need to be carefully designed and structured
  - Often requires diagrams to be drawn in particular way or authoring environment
  - Need for specialist software to access and interact with diagrams
  - Additional hurdles for both authors and readers

Goals

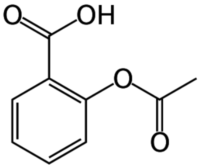
- Make regular teaching material accessible without the need to create new resources
- From inaccessible image to support for independent learning
- Source independence
  - Do not rely on the benevolent, educated author
- Tool independence
  - Do not require users to install/learn/use a specialist too.
- Provide a seamless user experience without/very little interface
- (Ideally) accessible with all browsers, screen readers
- Use standard web technology (HTML5, SVG, JavaScript)
- Support diverse material, for novices and experts alike

Examples

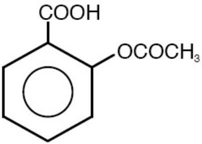
- Already Chemistry diagrams come in a variety of flavours depending on author preference and intended audience
- Different representations of Aspirin molecule.



Displayed formula.



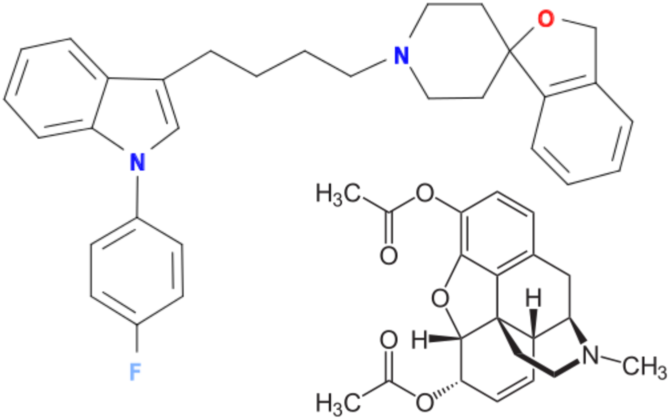
Skeletal formula.



Structural formula.

Examples

- Or somewhat more complex.



End-to-end Procedure from Images to Accessible Diagrams

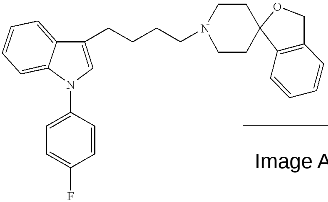
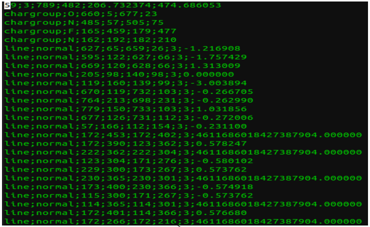
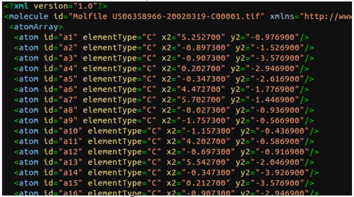


Image Analysis

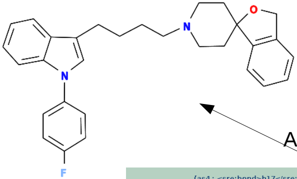


```
59;3;789;482;206.732374;474.686053
chargroup;O;660;5;677;23
chargroup;N;485;57;505;75
chargroup;F;165;459;179;477
chargroup;N;162;192;182;210
line;normal;627;65;659;26;3;-1.216908
line;normal;595;122;627;66;3;-1.757429
line;normal;669;120;628;66;3;1.313009
line;normal;205;98;140;98;3;0.000000
line;normal;119;160;139;99;3;-3.003894
line;normal;670;119;732;103;3;-0.266705
line;normal;764;213;698;231;3;-0.262990
line;normal;779;150;733;103;3;1.031856
line;normal;677;126;731;112;3;-0.272006
line;normal;57;166;112;154;3;-0.231100
line;normal;172;453;172;402;3;4611686018427387904.000000
line;normal;172;390;123;362;3;0.578247
line;normal;222;362;222;304;3;4611686018427387904.000000
line;normal;123;304;171;276;3;-0.580102
line;normal;229;300;173;267;3;0.573762
line;normal;230;365;230;301;3;4611686018427387904.000000
line;normal;173;400;230;366;3;-0.574918
line;normal;115;300;171;267;3;-0.573762
line;normal;114;365;114;301;3;4611686018427387904.000000
line;normal;172;401;114;366;3;0.576680
line;normal;172;266;172;216;3;4611686018427387904.000000
```

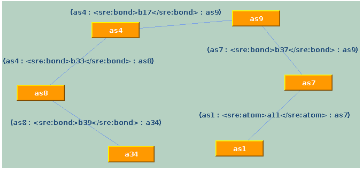

Image Recognition



```
<?xml version="1.0"?>
<molecule id="Molfile US06358966-20020319-C00001.tif" xmlns="http://www
 <atomArray>
  <atom id="a1" elementType="C" x2="5.252700" y2="-0.976900"/>
  <atom id="a2" elementType="C" x2="-0.897300" y2="-1.526900"/>
  <atom id="a3" elementType="C" x2="-0.907300" y2="-3.576900"/>
  <atom id="a4" elementType="C" x2="0.202700" y2="-2.946900"/>
  <atom id="a5" elementType="C" x2="-0.347300" y2="-2.616900"/>
  <atom id="a6" elementType="C" x2="4.472700" y2="-1.776900"/>
  <atom id="a7" elementType="C" x2="5.702700" y2="-1.446900"/>
  <atom id="a8" elementType="C" x2="-0.027300" y2="-0.936900"/>
  <atom id="a9" elementType="C" x2="-1.757300" y2="-0.566900"/>
  <atom id="a10" elementType="C" x2="-1.157300" y2="-0.436900"/>
  <atom id="a11" elementType="C" x2="4.202700" y2="-0.586900"/>
  <atom id="a12" elementType="C" x2="-0.697300" y2="-0.916900"/>
  <atom id="a13" elementType="C" x2="5.542700" y2="-2.046900"/>
  <atom id="a14" elementType="C" x2="-0.347300" y2="-3.926900"/>
  <atom id="a15" elementType="C" x2="0.212700" y2="-3.576900"/>
  <atom id="a16" elementType="C" x2="-0.907300" y2="-2.946900"/>
```

Accessible Rendering



Semantic Enrichment

Procedure

Input: A bitmap image of a molecule diagram

1. Image analysis and segmentation
2. Diagram recognition
3. Generation of annotated SVG
4. Semantic enrichment
5. Accessible diagram via browser front-end

Image Segmentation

Based on system we initially implemented for diagram recognition on patent databases

- Initial pre-processing: Binarisation, noise reduction...
- Connected component extraction and labelling
- Optical Character recognition and removal
- Thinning and smoothing with Douglas-Peuker
- Separation of bond elements
  - Walk skeleton diagram structure
  - Identify and break junction points
- Result is a set of geometric primitives:
  Character groups, lines, bold lines, circles, triangles

Diagram Recognition

- Rule based system
- Rewrites bag of geometric primitives into a graph representation
- Example:
  1. Let l_1, l_2 be distinct line segments of a minimum length.
  2. If l_1 is nearly parallel to and in a neighbourhood of l_2.
  3. No other line segment is nearly parallel to l_1 or l_2.

  $\Rightarrow$ Then (l_1, l_2) form a double bond.

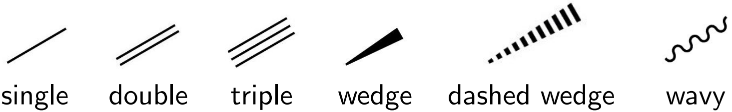


Diagram Recognition (ctd.)

- Not straight forward mapping of primitives to graph elements
- Also rewriting of primitives into other primitives
- Example of implicitly given carbon atoms

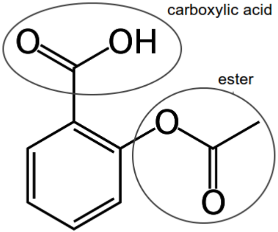
- Result is a Chemical Markup File (CML or MOL)
- But it is still a "flat" representation of a molecule

Annotated SVG Generation

- Many solutions for generating SVG from chemical markup
- But they only draw!
- And in the process destroy any structure or chemical knowledge
- Build our own SVG generator with emphasis on
  - Grouping meaningful units together (e.g., double bonds)
  - Retaining names given to components in the chemical markup (IDs of atoms, bonds, etc.)
- Result annotated and grouped SVG

Semantic Enrichment

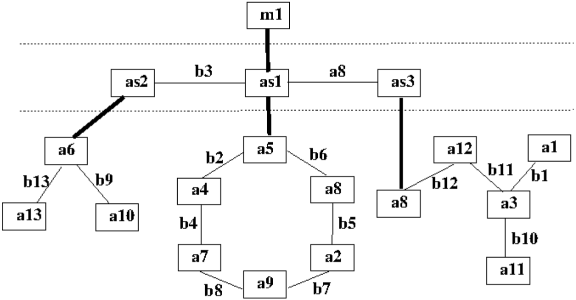
- Take basic chemical markup: Enrich it with derived knowledge and structure it accordingly
- Uses some cheminformatics algorithms from the CDK
- Detect major building blocks of the molecule
  - Aliphatic chains
  - Ring systems: Isolated and fused
  - Functional groups



- Order blocks and atoms by chemical conventions

Abstraction Graph

- Represent molecule as multi-layered graph
- 3-4 layers of abstraction



- Molecule
- Block
- Atom

- Additional layer in case of fused ring systems

Generating Descriptions

- Generate low level descriptions for atoms, bonds, and positions.
- High level descriptions for block elements:
  - Expert descriptions: Automatically name chemical compounds via ChemSpider and Cactus webservices
  - Basic descriptions via atoms, bonds, and substitutions
- Add abstraction graph and descriptions to original CML representation.

- Result semantically enriched CML File

Accessibility Support

- Graph structure can serve as the bases for interacting with the molecule
- Enables hierarchical exploration of molecule and its components
- Very simple navigation model: down/up, right/left
- Screen Reader Support:
  - Generate speech output from CML annotations on different levels
  - Display of speech output using subtitling
- Low Vision/Learning Disability Support:
  - Highlighting of inspected components
  - Optional zooming and magnification of components
  - Changing contrast, colour configurations

Browser Front-end

Generic browser front-end using standard web technology:

- Ajax service to import
  - annotated SVG
  - enriched CML as XML object
- Some JavaScript to tie it all together.
- WAI-ARIA and CSS to implement interactive exploration

Browser Front-end (Implementation)

- Container element for SVG and CML/XML document
  - `role` application with an appropriate `aria-label`
  - Both SVG and XML are `aria-hidden`
- Speech output for screen readers
  - Computation of speech string from CML components
  - Updating content of an `assertive` `aria-live` region
- Magnification and highlighting of explored substructures
  - Zooming by changing SVG view port
  - Highlighting by changing CSS properties
- Low vision and dyslexia support
  - Contrast changes by changing CSS properties

Browser Front-end (User Experience)

- Navigatable molecule is announced and can be entered on key
- Walk the CML structure using arrow keys
- Additional keys for special functionality (e.g., change of contrast)
- Execute visual and audio effects on the SVG
- Give the reader the feel of interacting with the diagram

User Feedback and Testing

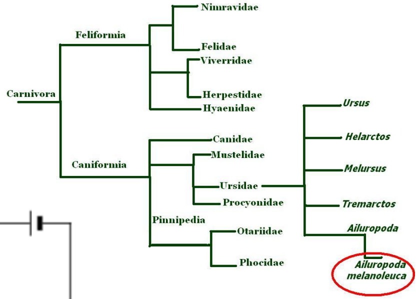
Ongoing stake holder involvement throughout development

- input from blind chemist (Duncan Bell)
- explanations tested in regular classroom teaching
- “Phone-experiments” with chemistry researchers
- “Molimod testing” with students at various levels in specialist college (NCW)
- Low vision support testing with A-level students
- Testing with educators for visually impaired children.

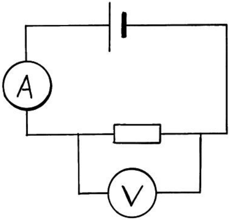
Current Work

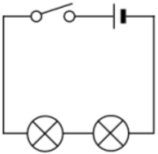
- Implementation of Touch Events
- Tactile diagrams, 3D printing, Localisation

- Extension to Biology (phylogenetic trees)



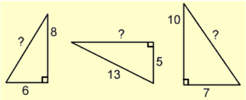
- Physics (circuit diagrams)

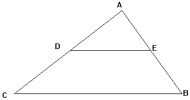


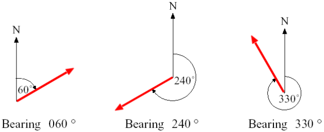


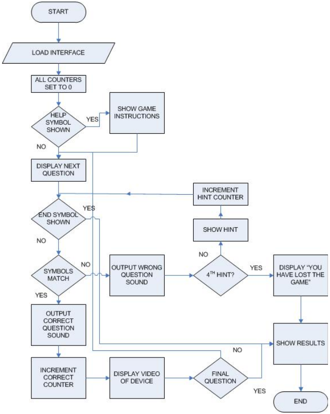
Future Work

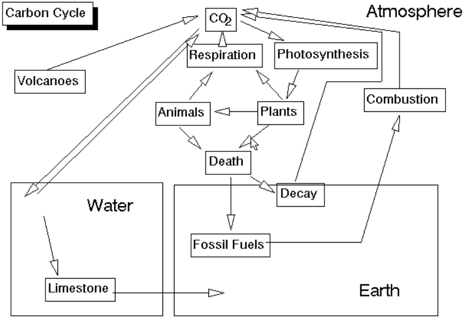
- Other STEM subjects: Maths (geometry, bearings), Biology (systems diagrams), Computer Science (flow charts)











Conclusions

- End-to-end procedure from images to accessible diagrams
  - WAI-ARIA needs to be expanded and become more flexible
  - Example: current work of the W3C SVG-4-A11Y task force
- No need to rely on author cooperation
- Integrates seamlessly without need for bespoke tools
- Works with most combinations of platforms, browsers, screen readers
  - Standardised interfaces in the Assistive Technology Ecosystem: screen readers, magnifiers, braille displays, ...
- Demo of web front end

  `http://progressiveaccess.com/chemistry`

Acknowledgements: Mark G. Lee, Sandy Wilkinson, Duncan Bell, Peter Murray-Rust, Egon Willighagen, John May, Noureddin Sadawi, John Saxon, students and teachers of NCW