Assistive Technologies for STEM Subjects

—

From Bitmap Graphics to Fully Accessible Chemical Diagrams

Volker Sorge

School of Computer Science
University of Birmingham
www.cs.bham.ac.uk/~vxs

Progressive Accessibility Solutions Ltd.
Birmingham, UK
progressiveaccess.com

Abstract

Diagrams are an important means of conveying information in STEM subjects starting as early as secondary school. However, even in electronic teaching material, diagrams are commonly given in standard image formats leaving them inaccessible for visually impaired learners.

In my work I aim to provide support for making STEM diagrams accessible, without the need for specialist tools for authoring or reading accessible diagrams. Instead I employ image analysis to recognise diagrams, semantic enrichment to derive detailed information on their content, and by regenerating them in a format that makes them amenable to assistive technology. Web browsing software allows readers to interactively engage with diagrams by exploring them step-wise and on different layers, enabling aural rendering of diagrams and their individual components together with highlighting and magnification to assist readers with low vision or learning difficulties.

The ideas have been successfully implemented and tested for chemical diagrams, building on open standards, supporting a number of computing platforms, browsers, and screen readers. However, the approach is generic, in that it is extensible to diagrams in other STEM subjects.

1 Introduction

Teaching in STEM subjects (i.e., Sciences, Technology, Engineering and Mathematics) often relies on the effective use of graphical depiction of important concepts. Images and diagrams are everywhere; not just to illustrate examples, but often they are the very subject of study themselves. For instance, consider diagrams of geometric shapes in mathematics or images of atoms and bonds in chemistry. Providing visually impaired learners access to this information similar to their sighted peers is still a major obstacle to realise fully inclusive teaching environments. In particular, mainstream schools or colleges do not necessarily have access to specialist equipment like embossers for tactile graphics or high quality magnification, leaving visually impaired students at a serious disadvantage.

The majority of these users rely on software-based assistive technology such as screen readers and magnifiers, which are mainly geared towards handling textual information and generally fail for diagrams. Since diagrams are usually given in standard bitmap image formats such as gif, png, and jpeg, they are effectively nothing more than rectangles of meaningless pixels, leaving them inaccessible to screen readers, since even text contained in the graphic is just a collection of pixels, which can not be voiced. Magnification tools also struggle with bitmaps since resolution does not increase proportionally with magnification, leading to a loss of image quality, which, in practice, renders most diagrams completely inaccessible to visually impaired users.

Even when alternative text descriptions are available, they in no way compare with the richness of information provided by even relatively simple diagrams. It is generally not sufficient to describe STEM diagrams superficially, as students need to know precisely what the diagram depicts. Moreover, diagrams are often too complex to grasp from a single description, but need to be explained step-wise, giving first an overview of what is depicted before diving deeper into single components.

While there have been a number of approaches to make scientific diagrams accessible, using a variety of means such as sonification, touch exploration and haptic feedback [2, 3, 4], these approaches are mostly suitable for relatively simple structures with homogeneous layouts, and can not be easily

adapted for semantically richer diagrams such as geometric figures, circuit diagrams or chemical diagrams. In particular for the latter a number of tools and environments have been developed to support visually impaired users in editing, reading and exploring molecules. Systems like Kekule [1], NavMol [5] and others [12], use chemical input languages or specialist input tools are used to generate molecule representations that allow users to explore molecules at different levels using keyboard driven navigation and speech output.

A drawback of all these approaches is that they need significant investment from authors to prepare diagrams, as well as forcing visually impaired readers to use specialist system they need to learn. I have therefore developed an end-to-end workflow for automatically making diagrams accessible, that eliminates these obstacles. Instead it employs (1) image analysis to recognise diagrams from any source and (2) semantic enrichment techniques to provide meaningful and detailed descriptions of diagrams that allows readers to interactively engage with them on multiple layers. It supports standard assistive technology solutions with a particular emphasis on screen reading and web technology. A successful proof-of-concept of the approach has been carried out with chemical diagrams [11].

2 Making Chemical Diagrams Accessible

The workflow to generate accessible diagrams from bitmap images combines four independent computational steps into a single software pipeline:

1. Image analysis recognises molecule diagrams regardless of authoring style and image source.
2. Semantic enrichment computes detailed and precise information on the depicted chemical molecule and its components.
3. Diagrams are faithfully reproduced in the Scalable Vector Graphics (SVG) format, which is amenable to scalable display and interaction in a web browser as well as offers facilities to attach the rich semantic information.
4. A navigation model on the SVG allows users to explore diagrams interactively while supporting speech output, synchronised highlighting and magnification in any ordinary web browser.

2.1 Image Analysis

The foundation of our approach is a robust and flexible image analysis mechanism that allows for the initial recognition of a diagram from any bitmap image. It is based on our previous work [10], which has shown itself superior in a number of international recognition competitions [9, 8]. It divides the image analysis into the following two distinct steps, one generic and one domain dependent, which makes the process easily extensible to other STEM diagram types:

Image Segmentation decomposes diagrams into a set of geometric primitives.

Diagram Recognition uses domain information to assemble a basic representation of the diagram.

In the *Image Segmentation* step an image is vectorised in order to segment it into the main constituents making up the diagram, resulting in a set of distinct primitives like lines, circles, solid triangles, arcs, or character groups together with their geometric location in the original image. The process consists of a number of steps, which are briefly summarised as follows: An initial preprocessing step binarises the image, and reduces noise by removing artefacts possibly introduced by image capturing techniques such as scanning. This yields a binary image, that is, one containing only black and white pixels, from which all the connected black components are extracted and labelled. In the next step, optical character recognition is performed by applying an OCR engine to the set of connected components and removing those recognised as characters from the image. This results in a skeleton diagram, to which thinning and smoothing techniques are applied to enable the identification and breaking of junction points to finally decompose the diagram into its geometric primitives.

Note that the image segmentation is fully generic, that is, it is independent of the actual type of diagrams analysed. Only the subsequent *Diagram Recognition* phase is based on knowledge about the

diagram domain. This recognition task is performed by a rule engine, in which largely disjoint rules are repeatedly applied to the initial set of geometric primitives, rewriting it into a graph representation of the given molecule diagram. The final graph structure serves as a basis from which efficient electronic representation formats can be generated.

Rules are defined in terms of preconditions and consequences. A rule is applicable if there exist geometric objects that satisfy its preconditions. Executing its consequence results in the removal of existing geometric objects and the addition of elements to the graph as well as possibly the addition of new geometric objects. In general, preconditions of different rules are mutually exclusive, and thus the order of rule application is irrelevant.

For the case of chemical diagrams, the rule engine is furnished with a set of rules that recognise components of molecules, such as different types of bonds. For example, two nearly parallel lines, with no other line segment close by, would be recognised as a double bond. The final graph structure is translated into the Chemical Markup Language (CML) [6], a standard chemical output format, that specifies molecules in terms of XML elements for atoms and bonds.

While the result of the image analysis is sufficient to reproduce the diagram and to distinguish molecules, the extracted information is still only sufficient for a flat representation, in the sense of describing single components of a diagram and their relationship to their direct neighbours. However, they lack sufficiently rich semantic that would be necessary to provide meaningful explanations of the diagram. For example, in the case of chemical diagrams, this amounts to an enumeration of all different bonds and which atoms they connect, however, there is no information on how atoms are clustered together into interesting compounds (e.g., rings) and their chemical meaning.

2.2 Semantic Enrichment

As the flat representation will only allow for very trivial description of components, the more challenging part is to enrich this representation with sufficient semantic meaning to allow the automatic generation of diagram descriptions that emulate human reading behaviour in its different facets, such as taking a casual glance at a drawing, getting an initial abstract overview of its components, before diving deeper into single components. In other words the semantic enrichment results in a representation that allows to generate descriptions on multiple layers, that is, not only for individual parts of the diagram but also for more complex components comprised of multiple parts as well as to give summary overviews.

This can be achieved by using graph analysis algorithms for clustering diagram components and defining meaningful levels of abstraction. For chemical diagrams, these clusters form rings, aliphatic chains, or functional groups, which in turn can be combined to more complex compounds such as multiple ring systems until they comprise the entire molecule. This segmentation already allows the generation of descriptions of the single compounds. In addition meaningful names for single components can be constructed. For chemical diagrams this is achieved by using online web services that generate common chemical names given a specification of a compound.

In a second step the identified components are assembled in a tree-like data structure where the level of a component in the tree corresponds to the level of abstraction it represents. Thus the root of the tree represents the entire diagram, while the leafs represent atomic components. Components on the same level of a tree are ordered by their importance, e.g., by molecular weight, and by neighbour relationship in the graph. Moreover, the single nodes of the tree are sub-graphs themselves and are related to neighbours in the tree via some relation in the graph and potentially they can also overlap in some components. Therefore, a level in the tree naturally defines a path through a single abstraction layer of the diagram. The resulting abstraction tree can then be exploited to compute descriptions of components with respect to their relation to neighbouring components as well as gives a means to generate summaries via sub-components.

2.3 Annotated SVG Generation

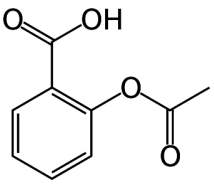
The generated CML representation of a molecule can serve as the basis to compute the corresponding diagram as SVG. Although there exist a number of solutions for this, these are exclusively geared towards rendering a diagram, discarding all chemical information in the process. That is, they will set all the geometric components, lines and characters, in a flat structure, losing information about bonds or atoms. As making a connection between the geometric component of the SVG and the bonds and atoms in the input CML file is important for the purpose of highlighting and magnification, we have implemented our own SVG renderer. It exploits SVG facilities to group elements together as well as to add attributes reflecting their chemical purpose and connecting them to their origins in CML.

2.4 Browser Front End

The abstraction tree generated by the semantic enrichment lends itself to a natural navigation model on the SVG diagram, which is realised via a simple browser front end. We employ AJAX functionality to import the annotated SVG and the enriched CML as an SVG+XML media type into the web page. This allows us to recreate a version of the abstraction tree inside the browser and to connect it to the corresponding components in the SVG. Some injected JavaScript code then enables interactive exploration of diagrams. The main idea is that a user can enter a diagram and interactively browse through its components on different levels and in different granularity. The components are presented to the reader by making descriptions available for aural rendering by a screen reader, highlighting and optionally magnifying focused parts of the molecule structure. All these operations are implemented browser, screen reader, and platform independent, exploiting HTML5 and WAI-ARIA standards.

3 Example

As an example we observe the single steps of the procedure with the recognition of a diagram for the Aspirin molecule, that is given on the right. The original file is a bitmap image in png format. Note that Aspirin is depicted as a so-called Skeletal formula, which simplifies the representation of molecules, by omitting most explicit hydrogen atoms, displaying neither their names nor their connecting bonds, as well as by not writing carbon atoms but only indicating them as unnamed end points of bonds.

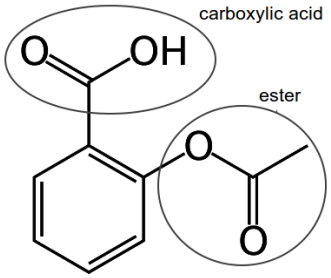


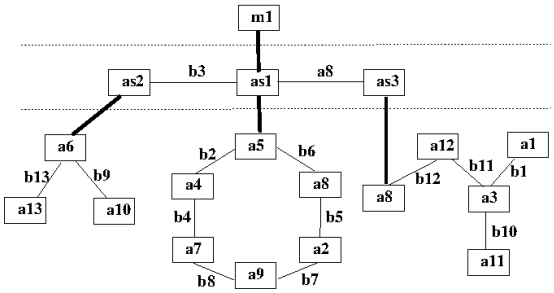
Step 1 Initially the image analysis proceeds with the image segmentation that results in a set of 22 geometric primitives: 18 lines and 4 character groups. These are given in the abbreviated list below on the left. The subsequent diagram recognition step then rewrites the bag of primitives into a CML representation. An extract of this initial CML structure is given below on the right.

```
60;4;336;279;188.992693;206.825861
chargroup;O;258;223;287;257
chargroup;O;195;114;224;148
chargroup;O;6;5;35;39
chargroup;OH;132;5;192;39
line;normal;82;62;85;56;4;4611686018427387904.0
line;normal;83;130;82;63;4;4611686018427387904.0
line;normal;83;276;145;240;4;-0.577093
line;normal;20;239;82;276;4;0.578389
line;normal;274;166;269;166;4;0.000000
line;normal;267;220;268;167;4;4611686018427387904.0
    ...
line;normal;81;62;34;36;4;0.575445
line;normal;86;56;127;32;4;-0.575723
line;normal;85;55;39;28;4;0.580944
```

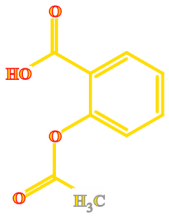
```
<molecule id="m1" xmlns="http://www.xml-cml.org/schema">
    <atomArray>
        <atom id="a1" elementType="C" x2="1.4301"
              y2="-0.6083" hydrogenCount="3"/>
        <atom id="a2" elementType="C" x2="-0.4599"
              y2="-1.6683" hydrogenCount="1"/>
        ....

    </atomArray>
    <bondArray>
        <bond id="b1" atomRefs2="a3 a1" order="S"/>
        <bond id="b2" atomRefs2="a5 a4" order="S"/>
        <bond id="b3" atomRefs2="a6 a5" order="S"/>
        <bond id="b4" atomRefs2="a7 a4" order="D"/>
        ....

    </bondArray>
</molecule>
```

Step 2 The semantic enrichment then proceeds to recognise the major components of the molecule. Aspirin effectively consists of three: a Benzene ring, a Carboxylic Acid and an Ester functional group. They are depicted below on the left. Once identified these components are combined and represented in the abstraction tree given below on the right.

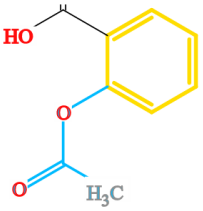




Step 3 The original CML structure allows us to generate an SVG of the Aspirin diagram that can be imported into any modern web browser. It can then be highlighted and magnified using CSS modifications and changes of the SVG viewport.
On the right is an image of the SVG diagram already highlighted on the top molecule level, that corresponds to the entire Aspirin molecule.



Step 4 Finally the browsing tool for molecule diagrams allows us to move vertically between the different levels of the graph presented in step 2 as well as horizontally between nodes on a single level of the graph. For example, one can move from the top level Aspirin molecule, down to the major component level and around this level. The image on the right depicts a move on the major component level, from the Benzene ring to the functional group Carboxylic Acid. This step is voiced as "Benzene ring with Carboxylic Acid at substitution 1".



4 Generalisation for Other STEM Diagrams

In order to generalise our procedure for generating accessible diagrams from chemistry to other STEM subjects, we need to separate the generic parts of the procedure from those domain specific to chemistry only. This has already been fully realised for the image analysis step, where we have a clear distinction between the generic image segmentation procedure and the diagram recognition step that is based on rules specific to molecule diagrams.

However, in future work we aim to extend this method to the semantic enrichment as well, achieving a similar separation into generic and domain specific part. The basic idea is to split the generation of the abstraction tree from the actual identification, ordering and naming of components. The latter could then again be define in a mainly rule based architecture.

Currently the initial graph is rewritten into the abstraction tree by hierarchically clustering components of the flat graph into increasingly complex, yet meaningful units. These are assembled in the abstraction tree such that each level of the tree represents a different level of abstraction of the diagram. Components on the same level are ordered by importance and connected by their neighbouring relationship in the diagram, thus naturally defining a path through a single abstraction layer of the diagram. This allows descriptions of components to be computed with respect to their relation to neighbouring components as well as summarising them via sub-components. This mechanism can effectively be based on generic algorithms like cycle detection or comparators.

On the other hand how components are composed, interpreted, ordered and described can be implemented in domain specific rules for different type of diagrams. For example, when ordering components in molecule diagrams we use chemical criteria such as molecular weight etc. Naturally, for other diagrams, like geometric drawings these ordering criteria would be different, for instance one could prefer interesting polygons over single points.

5 Conclusions

The current work on chemical diagrams already demonstrates that modern image analysis and semantic enrichment technology is strong enough to turn meaningless rectangles of pixels into fully accessible diagrams with detailed descriptions. Moreover, current web technology is already sufficient to provide rich interaction with SVG diagrams. And while improved WAI-ARIA standards and SVG features would still be beneficial, the current implementation already offers new levels of learner independence for visually impaired students. This allows us to make diagrams accessible without relying on the help of authors or the need for readers to use specialist software.

The approach is also not restricted to chemistry but extensible to diagrams in other STEM subjects. The current image analysis procedure is already clearly separated into a generic segmentation and a domain specific recognition part. We argue that a similar separation can be enforced for the semantic enrichment, splitting it into generic graph rewriting steps that are controlled by domain specific rules for identifying and naming components. If this can be fully achieved it would simplify the extension of our ideas to other STEM subjects like mathematics, physics and biology, thus furthering full inclusion of students with visual impairments into mainstream education.

References

[1] A. Brown, S. Pettifer, and R. Stevens. Evaluation of a non-visual molecule browser. *ACM SIGACCESS Accessibility and Computing*, 77-78:40–47, 2004.

[2] A. Brown, R. Stevens, and S. Pettifer. Audio representation of graphs: A quick look. In *Proc. of Auditory Displays*, 2006.

[3] L. Brown, S. Brewster, S. Ramloll, R. Burton, and B. Riedel. Design guidelines for audio presentation of graphs and tables. In *Proc. on Auditory Display*, 2003.

[4] R. Cohen, A. Meacham, and J. Skaff. Teaching graphs to visually impaired students using an active auditory interface. *ACM SIGCSE Bulletin*, 38(1):279–282, 2006.

[5] R. Fartaria, F. Pereira, V. Bonifácio, P. Mata, J. Aires-de Sousa, and A. Lobo. Navmol 2.0–a molecular structure navigator/editor for blind and visually impaired users. *Eur. J. Org. Chem.*, 2013(8):1415–1419, 2013.

[6] P. Murray-Rust and H. Rzepa. Chemical markup, xml and the world-wide web. 2. information objects and the cmldom. *J. Chem. Inf. Comput. Sci.*, 41(5):1113–1123, 2001.

[7] N. Sadawi, A. Sexton, and V. Sorge. Chemical structure recognition: a rule-based approach. In *Doc. Recognition & Retrieval XIX*, volume 8297. SPIE, 2012.

[8] N. Sadawi, A. Sexton, and V. Sorge. Molrec at CLEF 2012 — overview and analysis of results. In *CLEF Evaluation Labs*, 2012. `http://clef2012.org`.

[9] N. M. Sadawi, A. P. Sexton, and V. Sorge. Performance of MolRec at TREC 2011 — overview and analysis of results. In E. M. Voorhees and L. P. Buckland, editors, *The Twentieth Text REtrieval Conference Proceedings (TREC 2011)*, Gaithersburg, USA, Nov. 2011. National Institute of Standards and Technology (NIST). `http://trec.nist.gov/`.

[10] N. Sadawi, A. Sexton, and V. Sorge. Chemical structure recognition: a rule-based approach. In *Doc. Recognition & Retrieval XIX*, volume 8297. SPIE, 2012.

[11] V. Sorge, M. Lee, and S. Wilkinson. End-to-end solution for accessible chemical diagrams. In *Proceedings of the 12th Web for All Conference*, page 6. ACM, 2015.

[12] H. B. Wedler, S. R. Cohen, R. L. Davis, J. G. Harrison, M. R. Siebert, D. Willenbring, C. S. Hamann, J. T. Shaw, and D. J. Tantillo. Applied computational chemistry for the blind and visually impaired. *Journal of Chemical Education*, 89(11):1400–1404, 2012.