

# GraSSML: Accessible Smart Schematic Diagrams for All

Zaineb Ben Fredj and David Duce

Department of Computing

Oxford Brookes University, UK

[zbenfredj@brookes.ac.uk](mailto:zbenfredj@brookes.ac.uk)

[daduce@brookes.ac.uk](mailto:daduce@brookes.ac.uk)

# Motivations

# Motivations

- Diagrams are a **powerful** way of conveying information
  - *“A diagram is worth a thousand words”*
- Widely used on the web
- Have made life much easier for “**most**” users
- **Except for** users with specific disabilities or users who work in environments where graphical representations are inappropriate (e.g. screen size, text only display)

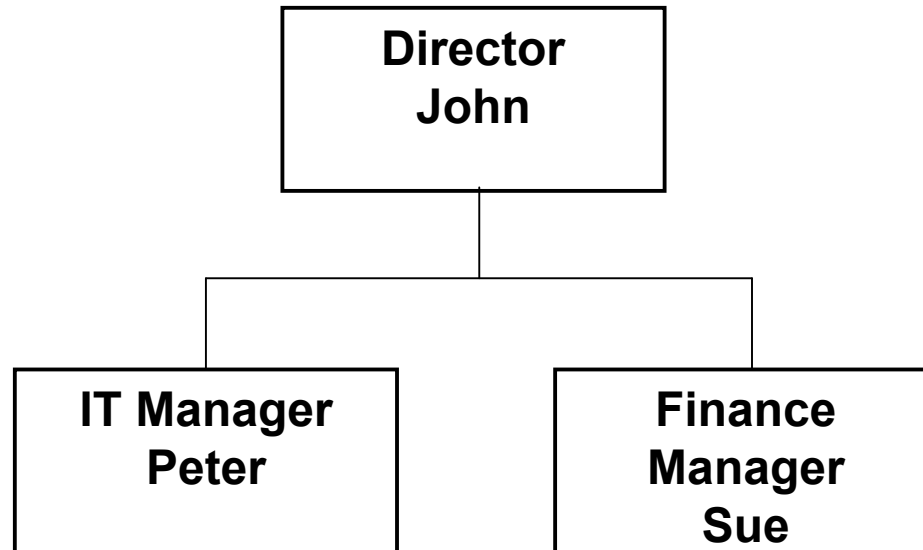
# The problem

# The problem

- Despite new web standards, the state of graphics on the web remains **limited**
- The web is still suffering from a **lack of techniques** to represent, generate, manipulate, adapt and access its diagrammatic content.
- Although a great deal of work has been done to address the accessibility problem of graphics, **some limitations remain.**

# Our approach

# A simple example



Do you see  
what I  
mean?

# Our approach

Let's start with

**what I mean**

instead of

what you see !

Generate “what you are going to see”

# Our approach

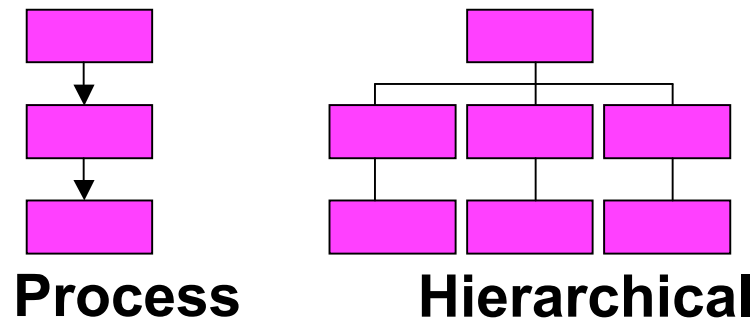
## Graphical Structure Semantic Markup Languages

(GraSSML) aims at:

- **facilitating** the **creation**, **modification** and **adaptation** of diagrams
- making the information “**behind**” the diagram “**part**” of the diagram
- make the semantics of the diagram machine-accessible
- **exploring** this information to generate alternative representations improving the **accessibility** of diagrams.

# Syntax and semantics of diagrams

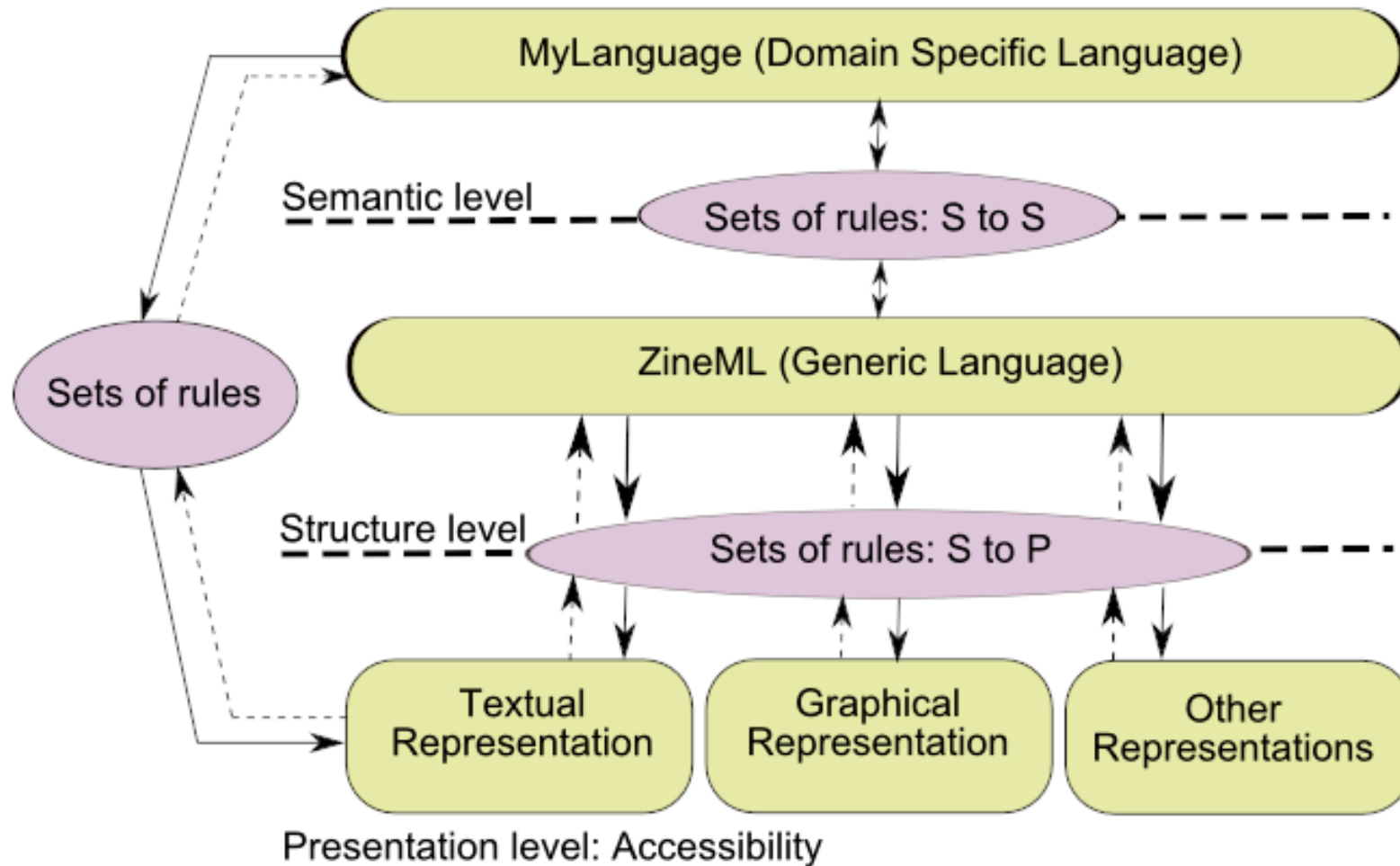
- **Very large** variety of diagrammatic notations and **no** universal visual conventions
- Start with well defined set of diagrams
  - Identify some **classes** of diagrams



- Identify domains with well defined conventions: **syntax** and **semantic rules**

# GraSSML Family of Languages

# GraSSML Family of Languages



# GraSSML Family of Languages: Semantic Level Example (Org\_Hierarchy)

- **Study of the domain and the class:** “Organisation chart, hierarchical diagrams”
  - **Concepts:** Employee, Director, Manager, It Manager, Finance Manager, Team Leader, Team Leader System, Team Leader Development, It Developer.
  - **Properties:** manages, reports to, name
- **Results of the study:**
  - The ontology OrgChart\_Ontology ([link to the ontology](#))
  - DS Markup Language “Org\_Hierarchy” of type hierarchical diagram.

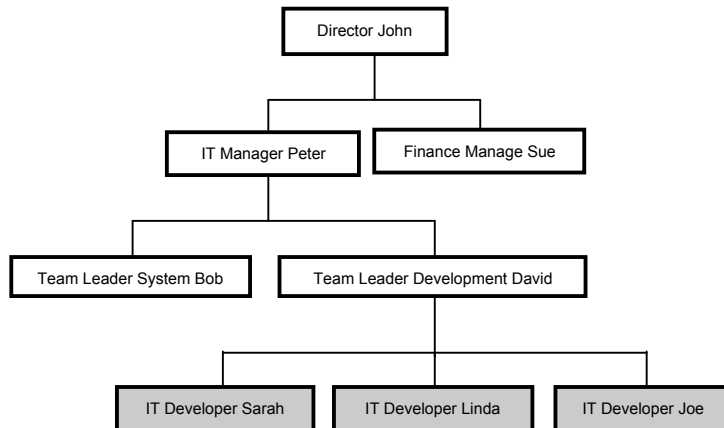
```
<Org_Hierarchy author="Z BEN FREDJ" date="February 2005" title="Organisation chart of my business">
  <Director id="D_John" name="John">
    <It_Manager id="Itm_Peter" name="Peter">
      <TeamLeader_System id="Tls_Bob" name="Bob"/>
      <TeamLeader_Development id="Tld_David" name="David">
        <It_Developer id="ItD_Sarah" name="Sarah"/>
        <It_Developer id="ItD_Linda" name="Linda"/>
        <It_Developer id="ItD_Joe" name="Joe"/>
      </TeamLeader_Development>
    </It_Manager>
    <Finance_Manager id="Fm_Sue" name="Sue"/>
  </Director>
</Org_Hierarchy>
```

# GraSSML Family of Languages: Structure Level: ZineML Example

```

<ZineML type="hierarchical" title="Organisation Chart of my business">
  <box> Director John
    <box> IT Manager Peter
      <box> Team Leader System Bob</box>
      <box> Team Leader Development David
        <box>IT Developer Sarah</box>
        <box>IT Developer Linda</box>
        <box>IT Developer Joe</box>
      </box>
    </box>
    <box> Finance Manager Sue </box>
  </box>
</ZineML>

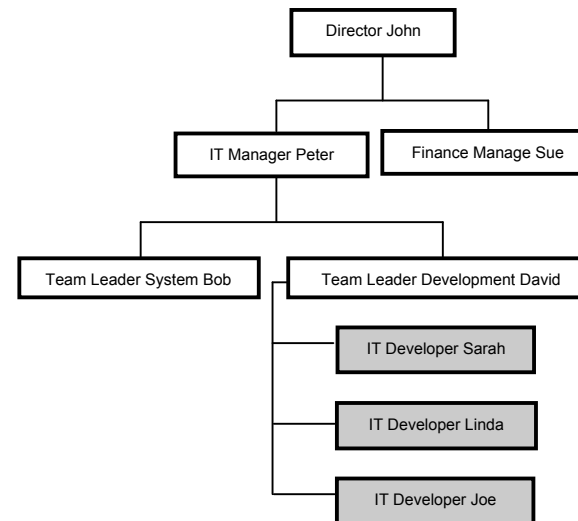
```



```

<ZineML type="hierarchical" title="Organisation Chart of my business">
  <box> Director John
    <box> IT Manager Peter
      <box>Team Leader System Bob</box>
      <box> Team Leader Development David
        <down> <box>IT Developer Sarah</box>
        <box>IT Developer Linda</box>
        <box>IT Developer Joe</box> </down>
      </box>
    </box>
    <box> Finance Manager Sue </box>
  </box>
</ZineML>

```



# GraSSML Family of Languages: Presentation Level (e.g. SVG, XHTML)

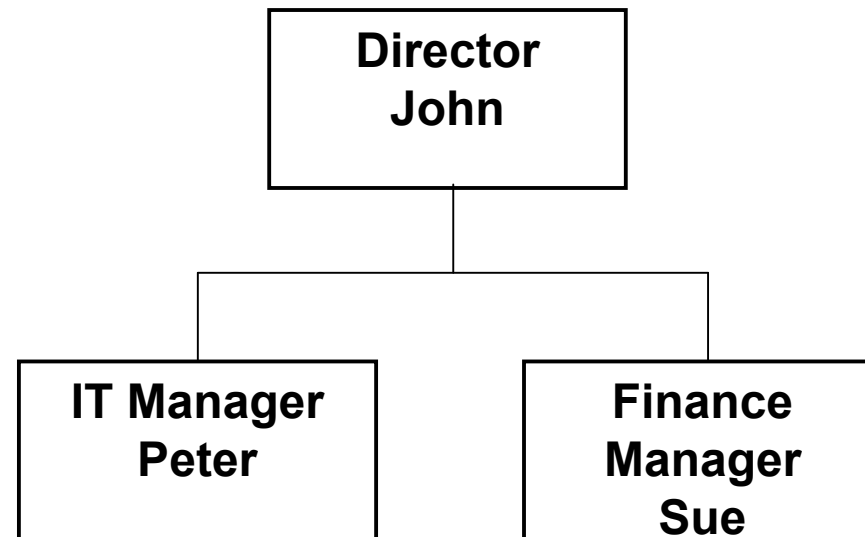
- Possible Alternative representations:
  - **Graphical**: SVG, possible interactive exploration of the diagram
  - **Textual**: verbalisation model of the structure and the semantics (XHTML)
  - **Query System**: query the structure and semantics of the diagram in novel ways.

# Related works

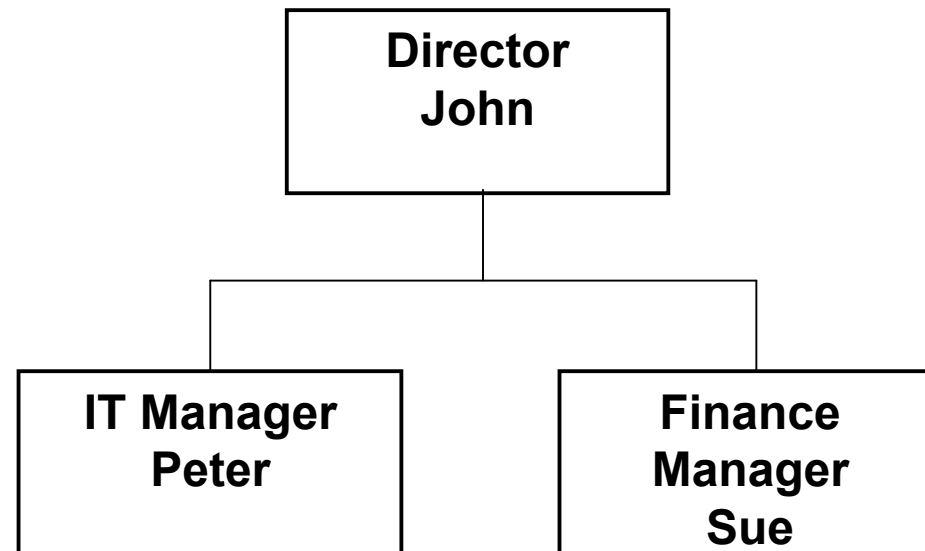
# “Bottom-up” approaches

- **Attempt to infer** alternative presentations from one given by an author (BIS, GUIB, TeDUB)
  - Advantages: presentation and navigation of the graphical information
  - Limitations: human intervention, important responsibility for the human involved, possibility of incomplete results

Do you see  
what  
I mean?



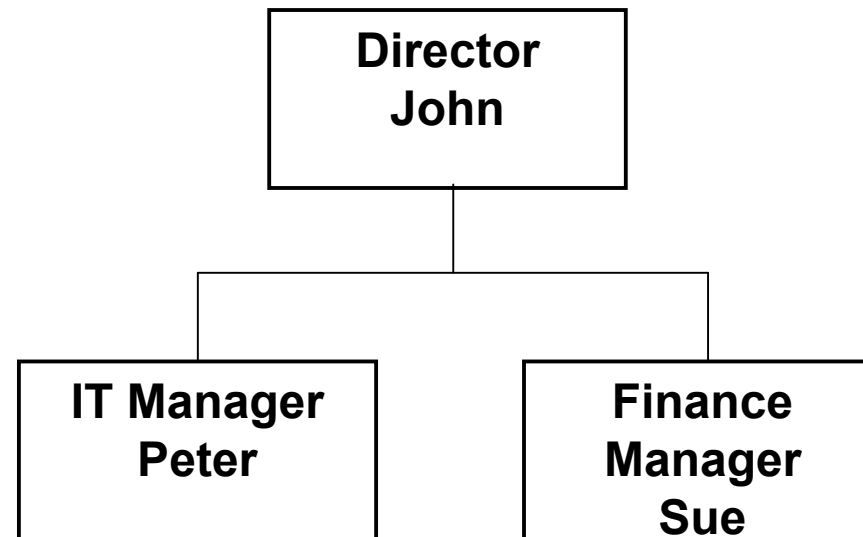
# SVG



# “Bottom-up” approaches

- **Attempt to embed** information into SVG file to overcome some of SVG’s limitations (CSVG, ViewPlus, Linearizer tool)
  - Advantages: explore the information behind the diagram, address some of SVG’s limitations successfully, permits a more flexible description of figures
  - Limitations: need too much efforts, tedious, time consuming, captures diagrams at the same low level as SVG

Do you see  
what  
I mean?



# Conclusion

# Conclusion

Start with the meaning  
instead of the  
marks on the screen

- **GraSSML** aims at defining higher-level diagram description languages which capture the structure and the semantics of a diagram and enable the generation of presentations in different modalities
- We address some issues which may have an important impact as an approach to the development of the **accessibility** of diagrams on the Web.
- “**bottom-up**” approaches offer many advantages but present some limitations mainly due to the lack of information “behind the diagram”
- **Not suggesting** that one should pursue only one approach  
BUT exploring our approach could offer great possibilities and clear benefits under certain circumstances

# Next steps

# Next steps

- Finalise the implementation of **GraSSML for the classes of process and hierarchical diagrams**
- Implement presentation and navigation tools
- Perform **user based experiments** to evaluate the usability, accessibility and limitations of the approach
- Consider **wider classes of diagrams**, with further development and experiments

Thanks for listening

Questions



Demo available

**Extra slides**

# SVG

## Advantages and accessibility benefits

- scalable, zoomable, searchable
- interoperability, Unicode support, CSS, XSL, DOM, wide tool support...
- “alternative equivalent”, style features, animation,...

## Limitations

- “final form presentation”
- Difficult at this level to handle the **resizing** and **positioning** in a complex diagram
- does not allow **flexible readjustment** of the layout
- little real scope for generating alternative presentations from the description at this level
- semantics of the diagram only “implicit”

# GraSSML Family of Languages:

## Structure Level: ZineML

- Aims at representing the structure of a diagram
- Aspire to **facilitate the creation and the modification** of diagrams (adjust positions and sizes semi-automatically)
- Proposes a set of basic shapes (extendable) for structured diagrams common to different fields
- Human readable
- Graphical representation is done in accordance with user predefined rule sets (Sets of rules S to P)
- **Domain independent**, but proposes some options to tailor the graphical representation specifically to a domain by expressing and applying rule sets when creating the diagram.
  - E.g. Hierarchy diagram represented from top to bottom (Rule sets: tags proposed by ZineML)

# GraSSML Family of Languages:

## Semantic Level (MyLanguage)

- XML Language aiming at capturing the **semantic** intent behind the diagram in a specific domain where clear conventions are followed
- Aims to capture the concepts and relationships between concepts that are to be expressed in pictorial or other representational form
- Aspire to be an intuitive **Domain Specific Language**
- Does not make any commitment to graphical representation but along with a particular MyLanguage, notational conventions are created (Sets of rules S to S) to govern the generation of a ZineML from MyLanguage
  - e.g. Director will be represented by a yellow box with his position and name centred*
- Process:
  - Study the domain and the class of diagrams in order to identify the concepts and relationships between the concepts the diagrams seek to represent
  - As a result of this study an **ontology** is created and a particular “MyLanguage” is created
  - Notational conventions defined for the graphical representation

# Implementation

